

HIGH SPEED TEXT SEARCH SYSTEM

HSTS SOFTWARE
LISTINGS

VOL. 4 OF 5

Diagnostics
Part I

STAT

NGA review(s) completed.

HSTS MASTER COMPUTER SOFTWARE LISTINGS

SL120100

VOLUME 4 of 5

Prepared for:

Central Intelligence Agency
Washington, DC 20505

☐ R80-016

March 1980

STAT

STAT

DIAGNOSTICS
PART 1

1 .TITLE--TMT-
2 .LIST MEB.
3
4
5
6
7
8
9

10 TERM DETECTOR MEMORY TESTS
11 MAIN MODULE
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

THIS MODULE EXECUTES MEMORY TEST CYCLES. ALL CONTROL INFORMATION FOR A TEST CYCLE IS OBTAINED FROM THE COMMAND LINE. TMT BUILDS TABLES AND SETS FLAGS IN ORDER THAT THERE NEED BE NO USER INTERVENTION DURING THE EXECUTION OF THE CYCLE. THE PROMPTS FOR COMMAND LINE INPUT AND GENERAL TMT ACTION ARE:

1. PROMPT FOR ALL TESTS, ALL MEMORIES, FULL RANGE. IF THE ANSWER IS 'Y', TMT BUILDS A COMPLETE MEMORY TEST CYCLING TABLE, THEN PROCEEDS TO 5. (FOR THE STRUCTURE OF THE TABLE, SEE BELOW REF. TABLE - CURRENT JUMP TABLE).
- IF THE ANSWER IS 'N', PROCEED TO 2 FOR MORE SELECTIVE PROMPTS.
2. PROMPT FOR MEMORIES TO BE TESTED. TMT SCANS THE COMMAND LINE RESPONSE AND SETS A FLAG FOR EACH MEMORY MNEMONIC IT FINDS.
3. PROMPT FOR MEMORY LIMITS. IF A MEMORY WAS SELECTED IN 2 (IE, ITS FLAG WAS SET), TMT PROMPTS FOR NUMERICAL VALUES WHICH DEFINE THAT PORTION OF A MEMORY ON WHICH THE TESTS ARE TO BE RUN. TMT PUTS THE MEMORY LIMITS FROM THE COMMAND LINE RESPONSE INTO A TABLE.
4. PROMPT FOR TESTS. TMT SCANS THE COMMAND LINE RESPONSE FOR TEST NUMBERS. THEN, FOR EACH MEMORY WHOSE FLAG IS UP, TMT BUILDS A ENTRY IN THE CURRENT JUMP TABLE (SEE BELOW).
5. PROMPT FOR LOOP ON TEST. TMT SETS A FLAG THAT DETERMINES WHETHER A TEST CYCLE WILL BE EXECUTED ONCE OR EXECUTED REPEATEDLY.
6. PROMPT FOR ERROR OPTIONS. TMT SETS FLAGS WHICH DETERMINE WHAT ACTION WILL BE TAKEN IN THE EVENT OF AN ERROR.
7. PUT OUT DIRECTIONS FOR STOPPING TEST. TMT ALLOWS THE TEST CYCLE TO BE INTERRUPTED BY AN UNSOLICITED CHARACTER INTERRUPT FROM THE TERMINAL.
8. TEST CYCLE BEGINS. NO MORE PROMPTING.

REFERENCE TABLE - CURRENT JUMP TABLE.

THE ACTUAL EXECUTION OF A TEST CYCLE DEPENDS UPON THE CONTENTS OF THE CURRENT JUMP TABLE. TMT FILLS IN THIS TABLE BY MOVING ENTRIES FROM THE REFERENCE TABLE DEPENDING UPON WHICH MEMORIES AND WHICH TESTS WERE SELECTED.

EACH TEST 1 - 12 HAS ITS OWN CONTROL ROUTINE IN TMT. THE REFERENCE TABLE ENTRIES ARE THE ADDRESSES OF THESE CONTROL ROUTINES. FOR EACH TEST, THE REFERENCE TABLE

58 : CONTAINS IN CONTIGUOUS POSITIONS THE CONTROL ROUTINE
59 : ADDRESS REPLICATED A NUMBER OF TIMES. THE NUMBER OF
60 : REPLICATIONS IS EQUAL TO THE NUMBER OF MEMORIES THAT
61 : CAN BE TESTED (IE. 6, THE VALUE OF THE EQUATE 'MEM').
62 : SO, FOR EXAMPLE, SINCE THERE ARE 6 MEMORIES, TEST 1'S
63 : CONTROL ROUTINE ADDRESS WILL BE REPEATED IN THE REFERENCE
64 : TABLE 6 TIMES:
65 :
66 : .WORD T1,T1,T1,T1,T1,T1
67 :
68 : EACH ADDRESS HERE IS A PLACE-HOLDER FOR A MEMORY. THAT IS,
69 : THE FIRST 'T1' (POSITION 0) IS A PLACE-HOLDER FOR FSA-A,
70 : THE SECOND 'T1' (POSITION 1) IS A PLACE-HOLDER FOR FSA-B,
71 : THE THIRD (POSITION 2) FOR FSA-C, THE FOURTH (POSITION 3)
72 : FOR THE INPUT BUFFER, THE FIFTH (POSITION 4) FOR THE
73 : OUTPUT BUFFER, AND THE SIXTH (POSITION 5) FOR THE BYTE
74 : TRANSLATOR. THE IDEA HERE IS THAT RATHER THAN HAVING SEPARATE
75 : CONTROL ROUTINES FOR EACH MEMORY FOR EACH TEST, TMT CAN
76 : MAKE USE OF THE POSITIONS OF ADDRESSES IN THE TABLE.
77 :
78 : THE FILLING OF THE CURRENT JUMP TABLE TAKES PLACE AS
79 : FOLLOWS: A UNIQUE FLAG IS SET IN A FLAG WORD FOR EACH
80 : MEMORY WHOSE MNEMONIC TMT ENCOUNTERS IN THE COMMAND
81 : LINE RESPONSE. THE PROMPT 'SELECT MEMORIES'. EACH
82 : POSITION IN THE FLAG WORD (0 - 5) CORRESPONDS TO A
83 : MEMORY PLACE HOLDER POSITION IN THE REF TABLE AND
84 : CURRENT JUMP TABLE. EG. THE FLAG FOR FSA-A IS IN
85 : POSITION 0 IN THE FLAG WORD AND THE PLACE-HOLDER
86 : POSITION FOR FSA-A IS 0 (SEE ABOVE). THEN FOR EACH TEST
87 : NUMBER TMT ENCOUNTERS IN THE COMMAND LINE RESPONSE TO
88 : THE PROMPT 'SELECT TEST(S)', TMT MOVES THE ADDRESS OF THAT
89 : TEST'S CONTROL ROUTINE FROM THE REF TABLE TO THE CURRENT
90 : JUMP TABLE DEPENDING UPON THE MEMORY FLAG SETTINGS. IE. TMT
91 : SCANS THE MEMORY FLAG WORD AND FOR EVERY BIT SET MOVES
92 : AN ADDRESS FROM THE REF TABLE TO THE CURRENT JUMP TABLE.
93 :
94 : EXAMPLE:
95 : IF TMT ENCOUNTERS A '1' IN THE COMMAND LINE RESPONSE TO
96 : 'SELECT TEST(S)' AND THE FLAGS FOR FSA-A AND THE OUTPUT
97 : BUFFER HAVE BEEN PREVIOUSLY SET, TMT WILL MOVE TEST 1'S
98 : CONTROL ROUTINE ADDRESS FROM THE REF TABLE TO THE CURRENT
99 : JUMP TABLE IN POSITIONS 0 AND 4 FOR TEST 1.
100 :
101 : REF TABLE:
102 : .WORD T1,T1,T1,T1,T1,T1
103 : .WORD T2,T2,T2,T2,T2,T2
104 :
105 : CURRENT JUMP TABLE:
106 : .WORD T1,0,0,0,T1,0
107 : .WORD 0,0,0,0,0,0
108 :
109 : MEMORY CYCLING
110 :
111 : THE CYCLE CONTROL ROUTINE MAINTAINS A POINTER TO THE
112 : CURRENT JUMP TABLE. THE CONTROL ROUTINE SCANS THE TABLE
113 : UNTIL IT FINDS A NON-ZERO ENTRY. IT DERIVES THE MEMORY
114 :

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

115      ;      TO BE TESTED FROM THE PLACE-HOLDING CHARACTERISTICS OF
116      ;      THE TABLE:
117      ;
118      ;      1. GETS THE POSITION OF THE CURRENT NON-ZERO ENTRY
119      ;      RELATIVE TO THE BEGINNING OF THE TABLE.
120      ;      2. DIVIDES THIS NUMBER BY THE NUMBER OF MEMORIES. THE
121      ;      REMAINDER GIVES THE PLACE-HOLDER VALUE.
122      ;
123      ;      FOR EXAMPLE, IF 'INPUT BUFFER' AND 'TEST 2' HAVE BEEN
124      ;      SELECTED, THE CURRENT JUMP TABLE WILL CONTAIN THE
125      ;      FOLLOWING INFORMATION:
126      ;
127      ;      .WORD 0,0,0,0,0,0
128      ;      .WORD 0,0,0,T2,0,0
129      ;
130      ;      THE FIRST NON-ZERO ENTRY IN THE TABLE IS AT OFFSET 9.
131      ;      THE REMAINDER FROM THE DIVISION OF 9 BY 6 (6 = NUMBER
132      ;      OF MEMORIES) IS 3. THE VALUE 3 IS THE PLACE-HOLDER
133      ;      VALUE FOR THE INPUT BUFFER.
134      ;
135      ;      THE CONTROL ROUTINE PASSES CONTROL TO THE TEST CONTROL
136      ;      ROUTINE WHOSE ADDRESS IS THE CURRENT NON-ZERO ENTRY IN
137      ;      THE CURRENT JUMP TABLE. THE CYCLE CONTROL ROUTINE PASSES
138      ;      THE REMAINDER FROM THE ABOVE DIVISION IN R0. THE TEST
139      ;      CONTROL ROUTINE USES THE CONTENTS OF R0 AS AN INDEX
140      ;      INTO A TABLE OF TMT SUB-MODULE MEMORY TEST ADDRESSES.
141      ;      THE TEST CONTROL ROUTINE IN TURN PASSES CONTROL TO THE
142      ;      ROUTINE IN THE SUB-MODULE THAT WILL EXECUTE THE TEST ON
143      ;      THE CORRECT MEMORY.
144      ;
145      ;      ALL SUB-MODULE ROUTINES ARE LOCATED IN THE MODULE TTEST.
146      ;
147      ;      ALL OF THESE ROUTINES RETURN TO THE TEST CONTROL ROUTINE
148      ;      THAT CALLED THEM. THE TEST CONTROL ROUTINES RETURN TO THE
149      ;      CYCLING ROUTINE WHICH SCANS THE CURRENT JUMP TABLE FOR THE
150      ;      NEXT NON-ZERO ENTRY.
151      ;
152      ;
153      ;      EXIT FROM THE PROGRAM DEPENDS UPON THE STATUS OF THE CURRENT
154      ;      JUMP TABLE, LOOP OPTIONS, HALT OPTIONS, OR TERMINAL INPUT.
155      ;
156      ;      JUMP TABLE EMPTY.      - EXIT.
157      ;      LOOP OPTION OFF.        - EXECUTE ONE TEST CYCLE.
158      ;      LOOP COUNT.             - EXECUTE A NUMBER OF CYCLES EQUAL TO
159      ;                                THE LOOP COUNT.
160      ;
161      ;      HALT OPTION ON.         - HALT AFTER ONE ERROR.
162      ;      COUNT + 'H'.           - PRINT A NUMBER OF MESSAGES EQUAL TO
163      ;                                THE COUNT AND HALT.
164      ;
165      ;      WHILE THE TESTS ARE RUNNING, THE ENTERING FROM THE TERMINAL
166      ;      OF ANY CHARACTER OTHER THAN W, C, P, OR T (THESE HAVE SPECIAL
167      ;      MEANINGS - SEE THE ROUTINE 'AST') STOPS THE TESTS IMMEDIATELY.
168      ;
169      ;
170      ;      MEMORY TESTS:
171      ;      TEST 01. WRITE MEMORY ADDRESS INTO ITSELF.

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
172.      ;      TEST-02. WRITE ZEROS..
173.      ;      TEST-03. WRITE ONES..
174.      ;      TEST-04. WRITE 125252..
175.      ;      TEST-05. WRITE/READ 146314 AND 031463
176.      ;      TEST-06. MEMORY CROSS-TALK TEST..
177.      ;      TEST-07. WRITE ADDRESS COMPLEMENT INTO ADDRESS..
178.      ;      TEST-08. WRITE 000377 AND 177400
179.      ;      TEST-09. SHIFT-BIT TEST..
180.      ;      TEST-10. WRITE USER SUPPLIED TEST PATTERN..
181.      ;      TEST-11. BIT MARCH TEST..
182.      ;      TEST-12. ADDRESSING TEST..
183.      ;
184.      ;
185.      ;      ASSEMBLY:      FROM [5.3].
186.      ;      MCR>MAC TMT,LP=IMQ4,TMT..
187.      ;
188.      ;      TASK BUILD:      ;ON-TERM DETECTOR-PACK..
189.      ;      TMT,TMT=TMT,TTEST..
190.      ;      /..
191.      ;      PAR=PAR14K..
192.      ;      TASK=...TMT..
193.      ;      ASG=TT0:1..
194.      ;      ASG=TT0:2..
195.      ;      /..
196.      ;      //
```

```

198      ;
199      ;
200      ; LOCAL DATA AREAS
201      ;
202      ;
203      ;
204      ; MCALL: Q10W$,Q10$,EXIT$,ABRT$,GCML$,GCMLB$,FSRSZ$,CLEF$,ASTX$,
205      ;
206      ;
207      ; LUN,TT = 1 ;LUN FOR TT0
208      ; EFN,1 = 1 ;EVENT FLAG FOR TT0
209      ; CMILUN = 2 ;LUN FOR GCML
210      ;
211      ; SETTINGS OR FLAG WORD 'BASE'
212      ;
213      ; LOOP = 1 ;LOOP FLAG
214      ; TEST6 = 2 ;TEST 6 IN EFFECT
215      ; TEST10 = 4 ;TEST 10 IN EFFECT
216      ; ALLTST = 10 ;ALL TESTS IN EFFECT
217      ; HALT = 20 ;HALT FLAG
218      ; ERROR = 40 ;ERROR HAS OCCURRED
219      ;
220      ; SETTINGS FOR FLAG WORD 'SELECT'
221      ;
222      ; A = 1 ;FSA - A
223      ; B = 2 ;FSA - B
224      ; C = 4 ;FSA - C
225      ; I = 10 ;INPUT BUFFER
226      ; O = 20 ;OUTPUT BUFFER
227      ; T = 40 ;BYTE TRANSLATOR
228      ;
229      ; NLIST BEX
230      MYSELF: .RAD50 /...TMT/
231      ASTURD: .WORD 0 ;ABORT THIS TASK
232      STAT: .BLKW 2 ;RECEIVER FOR CHAR FROM TERMINAL
233      ERWORD: .WORD 0 ;IO STATUS
234      ERLIM: .WORD 0 ;INFO/ERROR MESSAGE INDEX
235      BINWD: .WORD 0 ;MAX ERROR MESSAGES TO PRINT
236      UPPER: .WORD 0 ;VALUES CONVERTED FROM COMMAND LINE
237      LOWER: .WORD 0 ;WORK FIELD FOR UPPER MEMORY LIMITS
238      BASE: .WORD 0 ;WORK FIELD FOR LOWER MEMORY LIMITS
239      MT10: .WORD 0 ;GENERAL FLAG
240      T6FACT: .WORD 0 ;USER TEST PATTERN
241      GCMBUF: .BLKW 41 ;ADDRESS SKIP FACTOR FOR TEST 6
242      GCMLN: .WORD 0 ;COMMAND LINE BUFFER
243      GCMPNT: .WORD 0 ;COMMAND LINE LENGTH
244      PASSH: .WORD 0 ;POINTER TO COMMAND LINE
245      PASS: .WORD 1 ;HIGH WORD OF PASS COUNT
246      LOOPCT: .WORD 0 ;LOW WORD OF PASS COUNT
247      SELECT: .WORD 0 ;LOOP COUNT
248      MTPNT: .WORD 0 ;MEMORY SELECT FLAG WORD
249      MTCNT: .WORD 0 ;CURRENT JUMP TABLE POINTER
250      SETTBL: .WORD T$FSAA ;CURRENT JUMP TABLE COUNT
251      ; .WORD T$FSAB ;SELECT FSA-A
252      ; .WORD T$FSAC ;SELECT FSA-B
253      ;
254      ;

```

TABLE USED IN TRANSLATING FROM ASCII DECIMAL TO BINARY

```
255 000202. TROCT:..
256 000263' 000263' . =. .+61
257 000263 001 002 003 . .BYTE. 1,2,3,4,5,6,7,8,9,10,11,12.
258 000401' . =. TROCT+177
259 .
260 .
261 . STRING OF ALL TESTS (USED AS PSEUDO-COMMAND LINE)
262 000401 061 040 062 ALLSTR: .ASCII /1 2 3 4 5 6 7 8 9 11 12/
263 000027 STRLEN: =. -ALLSTR
264 .
265 .
266 .
267 . VALID MEMORY MNEMONICS AND MEMORY SELECT FLAG SETTINGS
268 000430 CTBL:
269 000430 101 115 .ASCII /AM/ ;FSA-A
270 000432 000001 .WORD. A.
271 000434 102 115 .ASCII /BM/ ;FSA-B
272 000436 000002 .WORD. B.
273 000440 103 115 .ASCII /CM/ ;FSA-C
274 000442 000004 .WORD. C.
275 000444 111 102 .ASCII /IB/ ;INPUT-BUFFER
276 000446 000010 .WORD. I.
277 000450 117 102 .ASCII /OB/ ;OUTPUT-BUFFER
278 000452 000020 .WORD. O.
279 000454 102 124 .ASCII /BT/ ;BYTE-TRANSLATOR
280 000456 000040 .WORD. T.
281 000006 CNUM: =. <-CTBL/4>
282 .
283 . MEMORY TEST CONTROL ROUTINE ADDRESSES (REFERENCE)
284 .
285 000460 MTREF:
286 000460 000064* 000064* 000064* .WORD. T1,T1,T1,T1,T1,T1
287 000474 000074* 000074* 000074* .WORD. T2,T2,T2,T2,T2,T2
288 000510 000110* 000110* 000110* .WORD. T3,T3,T3,T3,T3,T3
289 000524 000126* 000126* 000126* .WORD. T4,T4,T4,T4,T4,T4
290 000540 000144* 000144* 000144* .WORD. T5,T5,T5,T5,T5,T5
291 000554 000174* 000174* 000174* .WORD. T6,T6,T6,T6,T6,T6
292 000570 000200* 000200* 000200* .WORD. T7,T7,T7,T7,T7,T7
293 000604 000210* 000210* 000210* .WORD. T8,T8,T8,T8,T8,T8
294 000620 000240* 000240* 000240* .WORD. T9,T9,T9,T9,T9,T9
295 000634 000274* 000274* 000274* .WORD. T10,T10,T10,T10,T10,T10
296 000650 000300* 000300* 000300* .WORD. T11,T11,T11,T11,T11,T11
297 000664 000424* 000424* 000424* .WORD. T12,T12,T12,T12,T12,T12
298 .
299 . LOWER AND UPPER MEMORY LIMITS (REFERENCE)
300 .
301 000700 LIMREF:
302 000700 001777 .WORD. 1777 ;FSA-A-UPPER-LIMIT
303 000702 000000 .WORD. 0
304 000704 001777 .WORD. 1777 ;FSA-B-UPPER-LIMIT
305 000706 000000 .WORD. 0
306 000710 001777 .WORD. 1777 ;FSA-C-UPPER-LIMIT
307 000712 000000 .WORD. 0
308 000714 163776 .WORD. 163776 ;INPUT-BUFFER-UPPER-LIMIT
309 000716 160000 .WORD. 160000
310 000720 003776 .WORD. 3776 ;OUTPUT-BUFFER-UPPER-LIMIT
311 000722 000000 .WORD. 0
```

```
312.000724 000377 .WORD 255. ; BYTE TRANSLATOR UPPER LIMIT
313.000726 000000 .WORD 0
314 000014 LIMNUM: = <.-LIMREF>/2.
315
316 ;
317 ; MEMORY TEST ROUTINE ADDRESSES (CURRENT JUMP TABLE)
318
319 TESTS: = 12. ; NUMBER OF TESTS
320 000730 MEM: = 6. ; NUMBER OF MEMORIES
321 000730 MTSUB:
322 ;
323 ; .BLKW <TESTS*MEM>
324
325 ; LOWER AND UPPER MEMORY LIMITS (CURRENT TEST)
326 001150 CURLIM:
327 ;
328 ; .BLKW LIMNUM
329
330 ; SUB-MODULE MEMORY TEST ROUTINE ADDRESSES
331
332 STADDR: .WORD STUFF,STUFF,STUFF,STUFF,STUFF,STUFF
333 T1ADDR: .WORD T1F,T1F,T1F,T1I,T1O,T1T
334 T6ADDR: .WORD T6F,T6F,T6F,T6I,T6O,T6T
335 T7ADDR: .WORD T7F,T7F,T7F,T7I,T7O,T7T
336 TCDADD: .WORD T12FD,T12FD,T12FD,T12ID,T12OD,T12TD
337 TCUADD: .WORD T12FU,T12FU,T12FU,T12IU,T12OU,T12TU
338
339 ;
340 ; ERROR ROUTINE WORK AREAS
341
342 CKDATA: .WORD 0 ; TEST PATTERN
343 CK2: .WORD 0 ; TEST 12 READ TEST PATTERN
344 CK3: .WORD 0 ; TEST 12 WRITE TEST PATTERN
345 PREADD: .WORD 0 ; CURRENT MEMORY ADDRESS
346 ERRADD: .WORD 0 ; ADDRESS AT ERROR
347 ERW1: .WORD 0 ; NUMBER OF ERRORS
348 ERW2: .WORD 0 ; ERRONEOUS RESULTS FROM MEMORY - 1
349 ERW3: .WORD 0 ; - 2
350 ; - 3
351
352 ASWRK: .ASCII / /
353 TMSG: .ASCII /TEST/
354 PMSG: .ASCII /PASS/
355 FMSG: .ASCII /FAILING MEMORY: /
356 AMSG: .ASCII /ADDRESS: /
357 EMSG: .ASCII /EXPECTED: /
358 RMSG: .ASCII /RECEIVED: /
359 UNMSG: .ASCII /* * */
360
361 ;
362 ; MEMORY NAMES
363 FTBL:
364 .ASCII /FSA-A/
365 .ASCII /FSA-B/
366 .ASCII /FSA-C/
367 .ASCII /INPUT/
368 .ASCII /OUTPUT/
369 .ASCII /BTRANS/
```

```
369
370
371
372
373 001463      015      012
374 001465
375
376      000116
377
378
379
380
381
382
383
384
385 001603      000
386 001604      015      012      015
387 001610      124      105      123
388 001626      015      012      015
389 001632      124      105      123
390 001651      015      012
391 001653      105      116      124
392 001717      015      012      015
393 001723      115      105      115
394 001752      015      012      015
395 001760      124      105      122
396 002025      015      012      000
397 002030      015      012
398 002032      124      105      123
399 002122      015      012
400 002124      105      122      122
401 002157      015      012      000
402 002162      015      012
403 002164      111      116      126
404 002211      015      012
405 002213      111      116      126
406 002237      015      012
407 002241      111      116      126
408 002266      015      012
409 002270      111      116      126
410 002314      015      012
411 002316      111      114      114
412 002342      015      012
413 002344      111      116      126
414 002400      015      012
415 002402      111      116      126
416 002436      015      012
417 002440      111      116      126
418 002502      015      012
419 002504      111      116      126
420 002525      015      012
421 002527      105      116      124
422 002553      015      012
423 002555      114      117      117
424 002576      015      012
425 002600      105      116      124
```

PRINT LINE

PRINT: .BYTE 15,12 .PRECEDENCE PRINT LINE WITH CRLF

.NLIST MEB
.REPT 78
.BYTE 40
.ENDR
.LIST MEB

TABLE OF MESSAGES

.BYTE 0
.BYTE 15,12,15,12
.ASCIZ /TEST(S) ENDED/
.BYTE 15,12,15,12
.ASCIZ /TEST(S) HALTED/
.BYTE 15,12
.ASCIZ /ENTER ANY CHARACTER TO STOP TEST(S)/
.BYTE 15,12,15,12
.ASCIZ /MEMORY TEST(S) STARTED/
.BYTE 15,12,15,12,15,12
.ASCIZ /TERM DETECTOR MEMORY TEST DIAGNOSTICS/
.BYTE 15,12,0
.BYTE 15,12
.ASCIZ /TEST 6 INCOMPATABLE WITH MEMORY LIMITS. TEST DISCARDED/
.BYTE 15,12
.ASCIZ /ERROR: NO SELECTIONS. EXIT./
.BYTE 15,12,0
.BYTE 15,12
.ASCIZ /INVALID ERROR OPTION/
.BYTE 15,12
.ASCIZ /INVALID LOOP OPTION/
.BYTE 15,12
.ASCIZ /INVALID TEST PATTERN/
.BYTE 15,12
.ASCIZ /INVALID TEST NUMBER/
.BYTE 15,12
.ASCIZ /ILLEGAL ODD ADDRESS/
.BYTE 15,12
.ASCIZ /INVALID UPPER MEMORY LIMITS/
.BYTE 15,12
.ASCIZ /INVALID LOWER MEMORY LIMITS/
.BYTE 15,12
.ASCIZ /INVALID MEMORY OR BUFFER MNEMONIC/
.BYTE 15,12
.ASCIZ /INVALID RESPONSE/
.BYTE 15,12
.ASCIZ /ENTER ERROR CONTROL/
.BYTE 15,12
.ASCIZ /LOOP ON TEST(S)?/
.BYTE 15,12
.ASCIZ /ENTER PATTERN FOR TEST 10/

```

426 002632. 015 012 .BYTE 15,12.
427 002634 123 105 114 .ASCIZ /SELECT TEST/.
428 002650 015 012 .BYTE 15,12.
429 002652. 105 116 124 .ASCIZ /ENTER MEMORY LIMITS FOR BYTE TRANSLATOR/.
430 002722. 015 012 .BYTE 15,12.
431 002724 105 116 124 .ASCIZ /ENTER MEMORY LIMITS FOR OUTPUT BUFFER/.
432 002772. 015 012 .BYTE 15,12.
433 002774 105 116 124 .ASCIZ /ENTER MEMORY LIMITS FOR INPUT BUFFER/.
434 003041 015 012 .BYTE 15,12.
435 003043 105 116 124 .ASCIZ /ENTER MEMORY LIMITS FOR FSA-C/.
436 003101 015 012 .BYTE 15,12.
437 003103 105 116 124 .ASCIZ /ENTER MEMORY LIMITS FOR FSA-B/.
438 003141 015 012 .BYTE 15,12.
439 003143 105 116 124 .ASCIZ /ENTER MEMORY LIMITS FOR FSA-A/.
440 003201 015 012 .BYTE 15,12.
441 003203 123 105 114 .ASCIZ /SELECT MEMORY/.
442 003221 015 012 .BYTE 15,12.
443 003223 101 114 114 .ASCIZ /ALL TESTS, ALL MEMORIES, FULL RANGE?/.
444 003271 377 .ASCIZ .
445 .
446 .
447 003272. 120 101 123 PMSG2: .ASCIZ /PASS NUMBER/.
448 003272. 000014 . .-PMSG2.
449 .
450 .
451 003306 . 105 116 104 ENDOF: .ASCIZ /END OF PASS/.
452 003306 000014 . .-ENDOF.
453 .
454 .
455 .
456 .
457 .
458 .
459 .
460 .
461 .
462 003322. GCMBLK: GCMLB$ 2, GCMBUF, CMILUN.
003322. 002. .BYTE 2.
003364 002. .BYTE CMILUN.
003370 003510 . .WORD 65$
003376 001 . .BYTE 1
003341 0000 . .BYTE FD,TTY,FD,REC,FD,CCL.
003462. 000 . .BYTE 0
003463 047 . .BYTE GE,COM,GE,IND,GE,CLO,GE,SIZ.
003474 377 002. .BYTE -1,2
003476 003550 . .WORD 64$
003500 000032 . .WORD GCMBUF.
003502. 015 012 .ASCIZ <15><12>
003504 040 040 040 .ASCIZ />/.
003507 076 .
003516 012321 .RAD50 /CM1/.
003520 000000 .WORD 0
003522. 000000 .WORD 0
003524 012314 .RAD50 /CMD/.
003546 000120 .WORD 80.
463 003630 FRSZ$ 1

```



```

465      ;
466      ;
467      ; ENTER HERE
468      ;
469      ;
470      ;
471      ; PROMPT FOR ALL MEMORIES, ALL TESTS
472      ; START:
473      003630 004767 005240      JSR      PC,OUT1      ;ISSUE INFORMATION MESSAGE
474      003634 004767 005370      ALL:    JSR      PC,ALLSEL    ;ISSUE PROMPT
475      003640 103003              BCC      1$          ;NEED A RESPONSE
476      003642 004767 005302      JSR      PC,ERR2      ;INVALID RESPONSE
477      003646 000772              BR       ALL          ;PROMPT AGAIN
478      ;
479      003650 004767 004420      1$:      JSR      PC,FIND      ;LOCATE RESPONSE IN COMMAND LINE
480      003654 103003              BCC      2$          ;OK, VALIDATE RESPONSE
481      003656 004767 005266      JSR      PC,ERR2      ;
482      003662 000764              BR       ALL          ;PROMPT AGAIN
483      ;
484      ;
485      ; PARSE RESPONSE
486      003664 122711 000116      2$:      CMPB     #'N',(R1)      ;N = NO
487      003670 001435              BEQ      MTRTH          ;NOT ALL
488      003672 122711 000131      CMPB     #'Y',(R1)      ;Y = YES
489      003676 001403              BEQ      MOVE          ;OK, SET UP FOR ALL
490      003700 304767 005244      JSR      PC,ERR2      ;MUST BE Y OR N
491      003704 000753              BR       ALL          ;PROMPT AGAIN
492      ;
493      ; SET UP MEMORY LIMITS TABLE FOR CURRENT TEST
494      ; COPY REFERENCE TABLE TO CURRENT TABLE (IE, TEST
495      ; MEMORIES OVER THEIR FULL RANGE)
496      ;
497      003706 012700 000700      MOVE:    MOV      #LIMREF,R0      ;POINT TO REF TABLE
498      003712 012701 001150      MOV      #CURLIM,R1      ;POINT TO CURRENT TABLE
499      003716 012702 000014      MOV      #LIMNUM,R2      ;NUMBER OF WORDS TO MOVE
500      003722 012021              1$:      MOV      (R0)+,(R1)+
501      003724 005302              DEC      R2
502      003726 001375              BNE      1$
503      ;
504      ;
505      ; MOVE ALL OF MEMORY TEST REFERENCE TABLE TO CURRENT JUMP
506      ; TABLE
507      003730 012700 000460      MOV      #MTREF,R0      ;POINT TO REF TABLE
508      003734 012701 000730      MOV      #MTSUB,R1      ;POINT TO CURRENT TABLE
509      003740 012702 000110      MOV      #<TESTS*MEM>,R2 ;NUMBER OF WORDS
510      003744 012021              2$:      MOV      (R0)+,(R1)+
511      003746 005302              DEC      R2
512      003750 001375              BNE      2$
513      ;
514      ; JUMP TO LOOP SELECTION
515      ;
516      003752 052767 000010 174044 BIE      #ALLTST,BASE      ;SET FLAG FOR ALL TESTS
517      003760 000167 001046      JMP      LPRMPT

```

```

519
520
521
522
523
524
525 003764
526 003764 005067 174176
527 003770 004767 005230
528 003774 004767 004274
529 004000 103004
530 004002 052767 000077 174156
531 004010 000425
532
533
534
535
536
537 004012
538 004012 022700 000002
539 004016 001403
540 004020 004767 005120
541 004024 000757
542
543
544
545
546 004026 012700 000006
547 004032 012702 000430
548 004036 004767 004140
549 004042 103003
550 004044 004767 005074
551 004050 000745
552
553
554
555
556
557
558 004052 051167 174110
559 004056 004767 004212
560 004062 103353

```

```

;
;
; PROMPT FOR MEMORY SELECTION.
; IF RESPONSE IS <CR>, SELECT ALL MEMORIES.
;
;
MTRTN:
CLR. SELECT. ;CLEAR MEMORY SELECT FLAG.
JSR. PC, MEMSEL. ;PROMPT FOR MEMORY SELECTION
JSR. PC, FIND. ;FIND MEMORY MNEMONIC IN COMMAND LINE.
BCC. MEMTOP. ;A MNEMONIC WAS FOUND.
BIS. *A+B+C+I+O+T>. SELECT. ;SET ALL MEMORIES SELECTED.
BR. PLIM ;NOW PROMPT FOR MEMORY LIMITS.
;
; LOOP TO PROCESS MEMORY MNEMONICS (THERE MAY BE MORE
; THAN ONE IN THE COMMAND LINE). EG:
; >AM IB BT.
;
MEMTOP:
CMP. #2,R0 ;MNEMONIC LENGTH MUST BE 2.
BEQ. 2$ ;OK, CONTINUE.
JSR. PC, ERR3 ;INVALID MNEMONIC.
BR. MTRTN ;START OVER WITH PROMPT
;
; MATCH MEMORY MNEMONIC AGAINST A TABLE OF VALID MNEMONICS
; AND THEIR ASSOCIATED MEMORY FLAG SETTINGS.
;
2$: MOV. #NUM,R0 ;NUMBER OF VALID MEMORY MNEMONICS.
MOV. #CTBL,R2 ;POINT TO TABLE OF MNEMONICS
JSR. PC, SCAN. ;MATCH COMMAND LINE.
BCC. 3$ ;OK, CONTINUE.
JSR. PC, ERR3
BR. MTRTN ;START OVER.
;
; SUBRTN SCAN SETS R1 -> FLAG.
; LOOK FOR NEXT MNEMONIC IN COMMAND LINE. IF THERE IS
; ONE, PROCESS IT. IF THERE IS NONE, FALL THROUGH TO
; PROMPT FOR MEMORY LIMITS.
;
3$: BIS. (R1), SELECT. ;SET FLAG FOR MEMORY SELECTED.
JSR. PC, FIND. ;FIND NEXT MEMORY MNEMONIC IN COMMAND LINE.
BCC. MEMTOP. ;SOMETHING MORE IS THERE.

```

```

562.      ;
563.      ;
564.      ;      PROMPT FOR MEMORY LIMITS. READ AND VERIFY THEM.
565.      ;
566.      ;
567.      ;      SCAN THE MEMORY SELECT FLAG WORD FROM POSITION 0 TO
568.      ;      POSITION 5. FOR EVERY MEMORY WHOSE FLAG IS SET, CALL
569.      ;      SUBRTH 'LIMITS'.
570.      ;
571.      ;      PLIM:
572.      ;      MOV.      #A,R0      ; START TESTING WITH FSA-A
573.      ;      MOV.      #PMPTA,R1  ; POINT TO FIRST PROMPT MESSAGE
574.      ;      MOV.      #LIMREF,R2 ; POINT TO REFERENCE LIMITS TABLE
575.      ;      MOV.      #CURLIM,R3 ; POINT TO CURRENT LIMITS TABLE
576.      ;      MOV.      #MEM,R4   ; LOOP COUNT = NUMBER OF MEMORIES
577.      ;
578.      ;      1$:      BIT.      R0,SELECT  ; WAS MEMORY SELECTED
579.      ;      BEQ.      5$,      ; NO, BUMP POINTERS
580.      ;      2$:      MOV.      R0,-(SP)    ; SAVE TEST BIT
581.      ;      MOV.      R1,-(SP)    ; ADDR OF PROMPT MESSAGE
582.      ;      MOV.      (R2),-(SP)  ; MOVE UPPER REF LIMITS
583.      ;      MOV.      2(R2),-(SP) ; MOVE LOWER REF LIMITS
584.      ;      JSR.      PC,LIMITS
585.      ;      MOV.      (SP)+,2(R3)  ; MOVE IN CURRENT LOWER LIMITS
586.      ;      MOV.      (SP)+,(R3)  ; MOVE IN CURRENT UPPER LIMITS
587.      ;      MOV.      (SP)+,R1   ; RESTORE POINTER TO PROMPTS
588.      ;      MOV.      (SP)+,R0   ; RESTORE TEST BIT
589.      ;
590.      ;      INPUT BUFFER AND OUTPUT BUFFER INCREMENT THEIR
591.      ;      ADDRESSES BY 2. DO NOT ALLOW AN ODD ADDRESS IN
592.      ;      EITHER THEIR LOWER OR UPPER LIMITS.
593.      ;
594.      ;      3$:      BIT.      #1,R0      ; CHECKING INPUT BUFFER
595.      ;      BNE.      3$,      ; YES, DO NOT ALLOW ODD ADDRESS
596.      ;      BIT.      #0,R0      ; CHECKING OUTPUT BUFFER
597.      ;      BEQ.      5$,      ; NO, SKIP ODD ADDRESS CHECK
598.      ;      BIT.      #1,2(R3)   ; IS LOWER LIMIT ODD ?
599.      ;      BEQ.      4$,      ; NO, CONTINUE
600.      ;      JSR.      PC,ERR6
601.      ;      BR.      2$,      ; PROMPT AGAIN FOR LIMITS
602.      ;      4$:      BIT.      #1,(R3)   ; IS UPPER LIMIT ODD
603.      ;      BEQ.      5$,      ; NO, CONTINUE
604.      ;      JSR.      PC,ERR6
605.      ;      BR.      2$,      ; PROMPT AGAIN
606.      ;
607.      ;      5$:      DEC.      R4      ; FINISHED ?
608.      ;      BEQ.      ENDLIM  ; YES
609.      ;      ASL.      R0      ; SHIFT TO TEST NEXT BIT
610.      ;      SUB.      #4,R1     ; BACK UP PROMPT ADDR POINTER
611.      ;      ADD.      #4,R2     ; BUMP REF POINTER
612.      ;      ADD.      #4,R3     ; BUMP CURRENT POINTER
613.      ;      BR.      1$,      ; TEST NEXT

```

```

615
616
617
618
619
620
621
622
623
624
625
626
627 004240
628 004240 004767 004724
629 004244 004767 004024
630 004250 103021
631 004252 012700 000401
632 004256 012701 000032
633 004262 012702 000027
634 004266 112021
635 004270 005302
636 004272 001375
637 004274 012767 000027 173652
638 004302 012767 000032 173646
639 004310 004767 003760
640
641
642
643
644 004314
645 004314 022700 000002
646 004320 002416
647 004322 003007
648 004324 122721 000061
649 004330 001012
650 004332 111103
651 004334 062703 000012
652 004340 000401
653
654
655
656
657
658 004342 111103
659 004344 012704 000202
660 004350 060304
661 004352 111403
662 004354 001012
663
664
665
666
667 004356 004767 004542
668 004362 012700 000730
669 004366 012701 000110
670 004372 005020
671 004374 005301

```

```

;
;
; SELECT-TEST(S)
;
;
; PROMPT-FOR-TEST-NUMBERS: IF- THE- RESPONSE- IS- <CR>
; (CARRIAGE-RETURN-ONLY), MOVE-A-PSEUDO-COMMAND-LINE
; INTO- THE- COMMAND-LINE-BUFFER. THIS-PSEUDO-LINE
; CONSISTS-OF- THE- TEST-NUMBERS-FOR-ALL-TESTS-EXCEPT-
; TEST-10 (USER-PATTERN). PROCEED-TO-PROCESS-THIS-
; LINE-AS-THOUGH-IT-WAS-ENTERED-FROM- THE- TERMINAL.
;
;
; ENDLIM:
;
; JSR- PC,SELTST- ;PROMPT-TO-SELECT-A-TEST-
; JSR- PC,FIND- ;SCAN-COMMAND-LINE-FOR-A-TEST-NUMBER-
; BCC- SELOOP- ;FOUND-A-TEST-NUMBER-
; MOV- #ALLSTR,R0 ;POINT-TO-A-STRING-WITH-ALL-TESTS-
; MOV- #GCMBUF,R1 ;POINT-TO-COMMAND-BUFFER-
; MOV- #STRLEN,R2- ;GET-LENGTH-OF-STRING-
; 1$: MOV- (R0)+(R1)+ ;SIMULATE-ASKING-FOR-ALL-TESTS-
; DEC- R2-
; BNE- 1$
; MOV- #STRLEN,GCMLEN ;MOVE-LENGTH-OF-SIMULATED-STRING-
; MOV- #GCMBUF,GCMPT- ;INIT-POINTER-TO-COMMAND-LINE-
; JSR- PC,FIND- ;LOCATE-FIRST-TEST-NUMBER-IN-PSEUDO-LINE-
;
;
; PROCESS-ONE-TEST-NUMBER-AT-A-TIME. FIRST-VALIDATE-
; THE-NUMBER.
;
; SELOOP:
;
; CMP- #2,R0 ;TEST-NUMBERS-ARE-1-BYTE-EACH-
; BLT- SERR ;NO-GOOD
; BGT- 1$ ;SINGLE-DIGIT-PROCESS-IT-
; CMPB- #'1,(R1)+ ;TENS-DIGIT?
; BNE- SERR ;MUST-BE-1
; MOVB- (R1),R3 ;LOAD-1'S-DIGIT-
; ADD- #10,R3 ;ADD-VALUE-OF-TENS-DIGIT-
; BR- TRT- ;AND-CONTINUE-
;
;
; TRANSLATE-SINGLE-DIGIT-FROM-ASCII-DECIMAL-TO-BINARY.
; IF-A-ZERO-VALUE-IS-RETURNED-FROM-THE-TRANSLATION,
; THE-ASCII-CHARACTER-IS-INVALID.
;
;
; 1$: MOVB- (R1),R3 ;LOAD-TEST-NUMBER-(ASCII)
; TRT: MOV- #TROCT,R4 ;POINT-TO-TRANSLATE-TABLE-
; ADD- R3,R4 ;ADD-VALUE-OF-ASCII-NUMBER-
; MOVB- (R4),R3 ;MOVE-BINARY-VALUE-FROM-TABLE-
; BNE- ZREL ;SOMETHING-WAS-THERE-
;
;
; INVALID-CHARACTER. PUT-OUT-MESSAGE. CLEAR-CURRENT-JUMP-
; TABLE-AND-GO-BACK-TO-PROMPT-AGAIN.
;
;
; SERR: JSR- PC,ERR7
; MOV- #MTSUB,R0 ;POINT-TO-MSG-TABLE-
; MOV- *(TESTS*MEM),R1 ;LOAD-NUMBER-OF-WORDS-IN-TABLE-
; 3$: CLR- (R0)+ ;RESET-TABLE-
; DEC- R1

```

```

672 004376 001375      BNE 3$      ;TRY AGAIN.
673 004400 000717      BR  ENDLIM
674      ;
675      ; IF TEST NUMBER = 6 OR 10, SET A FLAG FOR
676      ; LATER ACTION.
677      ;
678 004402      ZREL:
679 004402 122703 000012  CMPB #10,R3      ;TEST 10 (USER TEST PATTERN)
680 004406 001003      BNE 1$      ;NO, DO NOT PROMPT
681 004410 052767 000004 173406 BIS #TEST10,BASE ;PROMPT FOR TEST PATTERN (LATER)
682 004416 122703 000006 1$:  CMPB #6,R3      ;TEST 6 (CROSS TALK)
683 004422 001003      BNE 2$      ;NO, DO NOT PROMPT
684 004424 052767 000002 173372 BIS #TEST6,BASE ;SET FLAG FOR LIMITS CHECK LATER
685      ;
686      ;
687      ; MAKE TEST NUMBER ZERO-RELATIVE, MULTIPLY THE ZERO-
688      ; RELATIVE TEST NUMBER BY THE NUMBER OF MEMORIES X 2.
689      ; TO GET A BYTE OFFSET INTO THE REFERENCE TABLE AND
690      ; CURRENT JUMP TABLE. FOR EXAMPLE, IF THE ASCII TEST NUMBER
691      ; WAS 2, THE ZERO-RELATIVE NUMBER IS 1. THIS NUMBER IS
692      ; MULTIPLIED BY 12 TO GET A BYTE OFFSET = 12.
693      ;
694      ;
695      ; ADD THE PRODUCT TO THE START ADDRESS OF THE CURRENT JUMP
696      ; TABLE AND PUT THE RESULT IN R0. ADD THE SAME PRODUCT TO
697      ; THE START ADDRESS OF THE REF TABLE AND PUT THE RESULT IN
698      ; R1. THE RESULTS ARE:
699      ;
700      ; CURRENT JUMP TABLE (ASSUMING TEST 2 SELECTED)
701      ; .WORD 0,0,0,0,0,0
702      ; .WORD 0,0,0,0,0,0
703      ;
704      ; REFERENCE TABLE
705      ; .WORD T1,T1,T1,T1,T1
706      ; .WORD T2,T2,T2,T2,T2
707      ;
708      ; R0 -> FIRST 0 IN THE SECOND LINE FOLLOWING "CURRENT JUMP TABLE"
709      ; R1 -> FIRST T2 IN THE REFERENCE TABLE.
710      ;
711 004432 005303 2$:  DEC R3      ;MAKE NUMBER ZERO-RELATIVE
712 004434 010301      MOV R3,R1      ;SYSTEM SUBRTN EXPECTS MULTIPLICAND IN R1
713 004436 012700 000014      MOV #<MEM*2>,R0 ;AND MULTIPLIER IN R0
714 004442 004767 000000G      JSR PC,$MUL
715 004446 010103      MOV R1,R3      ;MOVE PRODUCT TO R3
716 004450 012700 000730      MOV #MTSUB,R0 ;POINT TO JUMP TABLE
717 004454 060300      ADD R3,R0      ;ADD OFFSET
718 004456 012701 000460      MOV #MTREF,R1 ;TABLE WITH ALL SUBRTN ADDRESSES (REFERENCE)
719 004462 060301      ADD R3,R1      ;ADD OFFSET
720      ;
721      ;
722      ; DEPENDING UPON WHICH MEMORIES HAVE BEEN SELECTED FOR TESTING,
723      ; MOVE ADDRESSES OF ROUTINES THAT GOVERN TESTS FROM THE REFERENCE
724      ; TABLE TO THE JUMP TABLE (MTSUB). START TESTING THE MEMORY
725      ; SELECT FLAG AT POSITION 0 (FSA-A).
726      ;
727      ; PROCEEDING WITH THE ABOVE EXAMPLE ASSUMING IN ADDITION THAT
728      ; FSA-B WAS THE MEMORY SELECTED, THIS ROUTINE WOULD FILL
729      ; THE CURRENT JUMP TABLE IN THE FOLLOWING MANNER:

```

TMT-----MACRO: M1110 27-MAR-80 15:38 PAGE: 13-2.

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
729      ;
730      ;      .WORD 0.0.0.0.0
731      ;      .WORD 0.T2.0.0.0
732      ;
733 004464 012702 000001      MOV  #A,R2      ;START WITH FSA-A
734 004470 012703 000006      MOV  #MEM,R3      ;TOTAL NUMBER OF MEMORIES
735 004474 030267 173466      3$: BIT  R2,SELECT  ;WAS MEMORY SELECTED
736 004500 001401      BEQ  4$
737 004502 011110      MOV  (R1),(R0)      ;MOVE FROM REF TO CURRENT
738 004504 022120      4$: CMP  (R1)+,(R0)+      ;BUMP POINTERS
739 004506 006302      ASL  R2      ;SHIFT TO TEST NEXT BIT
740 004510 005303      DEC  R3      ;FINISHED?
741 004512 001370      BNE  3$
742      ;
743      ;      TRANSFERS BETWEEN REF TABLE AND CURRENT JUMP TABLE ARE
744      ;      COMPLETE FOR ONE TEST NUMBER. NOW SCAN THE COMMAND LINE
745      ;      FOR THE NEXT TEST NUMBER.
746      ;
747 004514 004767 003554      JSR  PC,FIND      ;FIND NEXT TEST REQUEST IN COMMAND LINE
748 004520 103275      BCC  SELOOP      ;PROCESS IT
```

```

750      ;
751      ;
752      ;
753      ;
754      ;
755      ;
756 004522 032767 000004 173274      BIT      #TEST10.BASE      ;WAS TEST 10 SELECTED
757 004530 001421                      BEQ      T6CHK      ;NO, SKIP ALL THIS
758 004532 004767 004426      PMPT10: JSR      PC,USRTST      ;PROMPT
759 004536 004767 003532      JSR      PC,FIND      ;FIND A NON-BLANK IN COMMAND LINE
760 004542 103003                      BCC      1$      ;OK, CONTINUE
761 004544 004767 004350      JSR      PC,ERR8
762 004550 000770                      BR      PMPT10
763 004552 004767 003634      1$:      JSR      PC,PACK      ;TRY AGAIN
764 004556 103003                      BCC      2$      ;CONVERT TEST PATTERN TO BINARY
765 004560 004767 004334      JSR      PC,ERR8      ;CONVERSION OK
766 004564 000762                      BR      PMPT10
767 004566 016767 173224 173232 2$:      MOV      BINWD,MT10      ;PUT PATTERN IN A SAFE PLACE
768      ;
769      ;
770      ;
771      ;
772      ;
773      ;
774      ;
775 004574      T6CHK:
776 004574 032767 000002 173222      BIT      #TEST6.BASE      ;WAS TEST 6 SELECTED
777 004602 001476                      BEQ      CHECK0      ;NO, SKIP AROUND
778 004604 005001                      CLR      R1      ;START PLACE-HOLDER VALUE = 0
779 004606 012700 001024      MOV      #MTSUB+(5*(MEM*2)),R0 ;POINT TO TEST 6 ADDRESSES
780 004612 012702 000006      MOV      #MEM,R2      ;LOOP COUNT = NUMBER OF MEMORIES
781      ;
782      ;
783      ;
784      ;
785      ;
786      ;
787      ;
788      ;
789      ;
790      ;
791      ;
792      ;
793      ;
794      ;
795      ;
796      ;
797      ;
798      ;
799      ;
800      ;
801      ;
802      ;
803      ;
804      ;
805      ;
806      ;

```

FINISHED WITH COMMAND LINE FOR TESTS.
 PROMPT FOR TEST PATTERN (IF TEST 10 WAS SELECTED).

CHECK FLAG TO SEE WHETHER TEST 6 WAS SELECTED. IF
 IT WAS, SET UP A POINTER TO THE CURRENT JUMP TABLE
 ENTRIES FOR TEST 6.

T6CHK:

FOR EACH MEMORY TO COME UNDER TEST 6, CHECK WHETHER
 THE MEMORY LIMITS ARE COMPATIBLE WITH THE TEST. TEST 6
 REQUIRES AT LEAST THREE MEMORY LOCATIONS IN ORDER TO
 WORK CORRECTLY.

HOW TEST 6 WORKS, RATIONALE BEHIND 3 LOCATION RULE.

TEST 6 CLEARS MEMORY FROM THE LOWER TO THE UPPER LIMIT.
 IT THEN WRITES ALL 1'S IN THE FIRST LOCATION AND EVERY
 OTHER LOCATION TO THE UPPER LIMIT. IT THEN READS ZEROS
 FROM THE LOCATIONS INTO WHICH IT DID NOT WRITE 1'S. IT
 THEN BUMPS THE LOWER LIMIT BY ONE MEMORY INCREMENT (VALUE
 VARIES DEPENDING UPON THE MEMORY). IT CLEARS MEMORY UP TO
 THE UPPER LIMIT. IT WRITES 1'S INTO THE NEW LOWER LIMIT
 AND EVERY OTHER LOCATION TO THE UPPER LIMIT. IT READS
 ZEROS FROM THE LOCATIONS INTO WHICH IT DID NOT WRITE
 1'S. THREE MEMORY LOCATIONS ARE THE MINIMUM ON WHICH
 TEST 6 CAN WORK.

WRITE 1'S INTO LOCATIONS 0 AND 2
 READ ZEROS FROM LOCATION 1
 177777

```

807      ;      000000
808      ;      177777
809      ;
810      ;      BUMP LOWER LIMIT TO 1
811      ;      CLEAR LOCATIONS 1 AND 2
812      ;      WRITE 1'S INTO LOCATION 1
813      ;      READ ZEROS FROM LOCATION 2
814      ;      177777
815      ;      177777
816      ;      000000
817      ;
818      ; *****
819      ;
820      ;
821      ;      CHECK EACH NON-ZERO TEST 6 ENTRY IN THE CURRENT JUMP TABLE
822      ;      FOR EACH NON-ZERO ENTRY, SET UP IN R4 THE MEMORY INCREMENT
823      ;      VALUE X 2 (IE, THE MINIMUM SPREAD FOR TEST 6). BASIC VALUES:
824      ;
825      ;      FSA-A 1
826      ;      FSA-B 1
827      ;      FSA-C 1
828      ;      INPUT 2
829      ;      OUTPUT 2
830      ;      BTRANS 1
831      ;
832 004616 005710 CHECK: TST (R0) ;TEST 6 SELECTED FOR THIS MEMORY
833 004620 001414 BEQ 2$ ;NO, SKIP CHECK
834 004622 022701 000005 CMP #5,R1 ;BYTE TRANSLATOR
835 004626 001003 BNE 1$ ;
836 004630 012704 000002 MOV #2,R4 ;VALUE X 2
837 004634 000416 BR 4$ ;
838 004636 022701 000004 1$: CMP #4,R1 ;OUTPUT BUFFER
839 004642 001003 BNE 2$ ;
840 004644 012704 000004 MOV #4,R4 ;VALUE X 2
841 004650 000410 BR 4$ ;
842 004652 022701 000003 2$: CMP #3,R1 ;INPUT BUFFER
843 004656 001003 BNE 3$ ;
844 004660 012704 000004 MOV #4,R4 ;VALUE X 2
845 004664 000402 BR 4$ ;
846 004666 012704 000002 3$: MOV #2,R4 ;VALUE X 2 FOR FSA'S
847      ;
848      ;      USE THE PLACE-HOLDER VALUE AS AN INDEX INTO THE CURRENT
849      ;      LIMITS TABLE (AFTER SHIFTING FOR DOUBLE-WORD OFFSET). THEN
850      ;      CHECK THE RELATIONSHIP BETWEEN LOWER AND UPPER LIMITS.
851      ;
852 004672 010103 4$: MOV R1,R3 ;SHIFT IN ANOTHER REG
853 004674 006303 ASL R3 ;
854 004676 006303 ASL R3 ;SHIFT FOR DOUBLE WORD OFFSET
855 004700 006304 001152' ADD CURLIM+2(R3),R4 ;GET LOWER LIMIT + MINIMUM INCR
856 004704 026304 001150' CMP CURLIM(R3),R4 ;IS UPPER LIMIT OK FOR TEST 6
857 004710 103026 BHJS 6$ ;YES, CONTINUE
858      ;
859      ;      FAILED CHECK. CLEAR THE ENTRY IN THE CURRENT JUMP TABLE
860      ;      AND REPORT TO THE CONSOLE. R1 = PLACE-HOLDER VALUE. THIS
861      ;      VALUE IS USED AS AN INDEX INTO A TABLE OF MEMORY NAMES.
862      ;      (AFTER MULTIPLYING THE VALUE BY 6, THE LENGTH OF EACH
863      ;      NAME).

```



```

864
865 004712 005010          CLR (R0)          ;CLEAR TEST 6 ADDRESS FROM CURRENT TABLE
866 004714 010046          MOV R0,-(SP)        ;SAVE POINTER
867 004716 010146          MOV R1,-(SP)        ;SAVE PLACE-HOLDER VALUE
868 004720 004767 004154    JSR PC,ERR12       ;PRINT GENERAL ERROR MESSAGE
869 004724 012700 000006    MOV #6,R0         ;LENGTH OF MEMORY NAMES
870 004730 004767 000000G   JSR PC,$MUL      ;GET OFFSET INTO MEMORY NAME TABLE (R0XR1)
871 004734 012700 001417    MOV #FTBL,R0      ;POINT TO MEMORY NAME TABLE
872 004740 012701 000006    MOV #6,R1         ;NUMBER OF CHARS IN NAME
873 004744 012705 001465    MOV *PRINT,R5     ;POINT TO PRINT LINE
874 004750 112025          5$: MOVB (R0)+,(R5)+ ;MOVE NAME TO PRINT LINE
875 004752 005301          DEC R1
876 004754 001375          BNE 5$
877 004756 004767 003720    JSR PC,CONSOL     ;WRITE MEMORY IN ERROR
878 004762 012601          MOV (SP)+,R1
879 004764 012600          MOV (SP)+,R0
880
881
882
883 004766 005201          : PREPARE TO CHECK NEXT TEST 6 ENTRY
884 004770 062700 000002    6$: INC R1         ;BUMP MEMORY PLACE-HOLDER VALUE
885 004774 005302          ADD #2,R0           ;POINT TO NEXT TEST 6 ADDRESS
886 004776 001307          DEC R2             ;SUB FROM LOOP COUNT
887
888
889
890
891
892 005000          : MAKE SURE THAT THE 'CURRENT JUMP TABLE' IS NOT EMPTY
893 005000 012700 000730    CHECK0: MOV #MTSUB,R0 ;POINT TO CURRENT TABLE
894 005004 012701 000110    MOV *MEM*TESTS>,R1 ;NUMBER OF TABLE ENTRIES
895 005010 005720          1$: TST (R0)+      ;IS A TABLE ENTRY PRESENT
896 005012 001007          BNE LPRMPT        ;YES, EXIT THIS ROUTINE
897 005014 005301          DEC R1            ;SUB FROM ROUTINE COUNT
898 005016 001374          BNE 1$           ;TRY NEXT POSITION
899 005020 004767 004060    JSR PC,ERR11     ;EXECUTION IMPOSSIBLE
900
901 005024          :
902 005024 012746          MOV (PC)+,-(SP)
903 005026 063          .BYTE 51,1
904 005030 104377          EMT <0<377>

```

```

903      ;
904      ;
905      ;      PROMPT FOR LOOP ON TEST
906      ;
907      ;
908      ;      RESPONSES:
909      ;      <CR>      - CARRIAGE RETURN LOOP ON TESTS
910      ;      Y      - YES LOOP ON TESTS
911      ;      N      - NO ONE MEMORY TEST CYCLE ONLY
912      ;      NUMERIC VALUE - NUMBER OF CYCLES TO EXECUTE
913      ;
914      005032      LPRMPT:
915      005032      004767      004122      JSR      PC,LPTST      ;PROMPT
916      005036      004767      003232      JSR      PC,FIND      ;FIND RESPONSE IN COMMAND LINE
917      005042      103004      BCC      1$      ;OK, RESPONSE FOUND
918      005044      052767      000001      172752      BIS      #LOOP,BASE      ;CR RESPONSE MEANS LOOP
919      005052      000442      BR      ERPRMT      ;AND CONTINUE
920      ;
921      005054      122711      000131      1$:      CMPB      #'Y,(R1)      ;YES - LOOP ON TESTS
922      005060      001004      BNE      2$      ;TRY 'N'
923      005062      052767      000001      172734      BIS      #LOOP,BASE      ;SET FLAG FOR LOOP
924      005070      000433      BR      ERPRMT      ;PROMPT FOR ERROR OPTIONS
925      005072      122711      000116      2$:      CMPB      #'N,(R1)      ;NO - DO NOT LOOP ON TESTS
926      005076      001004      BNE      3$      ;NO, TEST FOR LOOP COUNT
927      005100      042767      000001      172716      BIC      #LOOP,BASE      ;CLEAR LOOP FLAG
928      005106      000424      BR      ERPRMT
929      ;
930      ;      ASSUME THAT THERE IS AN ASCII DECIMAL VALUE IN THE COMMAND
931      ;      LINE. CONVERT IT TO BINARY AND STORE
932      ;
933      005110      060100      3$:      ADD      R1,R0      ;POINT 1 PAST STRING
934      005112      005200      INC      R0      ;BUMP FOR STUPID SYSTEM SUBRTN
935      005114      010046      MOV      R0,-(SP)      ;SAVE FOR LATER COMPARISON
936      005116      010100      MOV      R1,R0      ;MOVE POINTER TO R0 FOR SYSTEM SUBRTN
937      005120      004767      000000G      JSR      PC,$CDTB      ;CONVERT DECIMAL TO BINARY
938      005124      020026      CMP      R0,(SP)+      ;WHOLE STRING CONVERTED
939      005126      001403      BEQ      4$      ;YES, CONTINUE
940      005130      004767      003760      JSR      PC,ERR9
941      005134      000736      BR      LPRMPT      ;PROMPT AGAIN
942      005136      010167      173022      4$:      MOV      R1,LOOPCT      ;SAVE LOOP COUNT
943      005142      001003      BNE      5$
944      005144      004767      003744      JSR      PC,ERR9
945      005150      000730      BR      LPRMPT
946      005152      052767      000001      172644      5$:      BIS      #LOOP,BASE      ;SET LOOP FLAG

```

```

948
949
950
951
952
953
954
955
956
957
958
959
960 005160
961 005160 004767 003770
962 005164 004767 003104
963 005170 103445
964
965 005172 122711 000110
966 005176 001007
967 005200 052767 000020 172616
968 005206 012767 000002 172600
969 005214 000433
970
971
972
973
974 005216 000100
975 005220 005200
976 005222 010046
977 005224 010100
978 005226 004767 000000G
979 005232 020026
980 005234 001403
981 005236 004767 003646
982 005242 000746
983
984
985
986 005244 010167 172544
987 005250 005267 172540
988 005254 004767 003014
989 005260 103411
990 005262 122711 000110
991 005266 001403
992 005270 004767 003614
993 005274 000731
994
995 005276 052767 000020 172520

```

```

:
:
: PROMPT FOR ERROR OPTIONS.
:
:
: RESPONSES:
: <CR> - CARRIAGE RETURN. DISPLAY ALL ERRORS.
: H - HALT. STOP TESTS AFTER 1ST ERROR.
: NUMERIC VALUE - PRINT ONLY THIS NUMBER OF ERROR MESSAGES.
: BUT CONTINUE TESTS.
: N. VALUE + H - PRINT THIS NUMBER OF ERROR MESSAGES AND HALT.
:
: ERPRMT:
: JSR PC, EROPT ; PROMPT FOR OPTIONS.
: JSR PC, FIND ; FIND RESPONSE.
: BCS MTSET ; <CR> - DEFAULT TO "DISPLAY"
:
: CMPB #H, (R1) ; HALT AFTER 1 ERROR.
: BNE 1$ ; NO.
: BIS #HALT, BASE ; SET FLAG FOR HALT.
: MOV #2, ERLIM ; PRINT ONLY 1- ERROR MESSAGE.
: BR MTSET
:
: ASSUME AN ASCII DECIMAL VALUE IN THE COMMAND LINE. CONVERT
: AND STORE.
:
: 1$: ADD R1, R0 ; POINT PAST STRING IN COMMAND LINE.
: INC R0 ; BUMP FOR STUPID SYSTEM SUBRTN.
: MOV R0, -(SP) ; SAVE FOR LATER COMPARISON
: MOV R1, R0 ; PREPARE TO CONVERT.
: JSR PC, $CDTB.
: CMP R0, (SP)+ ; ENTIRE STRING CONVERTED.
: BEQ 2$ ; YES.
: JSR PC, ERR10
: BR ERPRMT ; TRY AGAIN.
:
: LOOK FOR AND "H" AFTER THE ASCII OCTAL VALUE.
:
: 2$: MOV R1, ERLIM ; NUMBER OF ERROR MSGS TO PRINT
: INC ERLIM ; ADJUST FOR PRE-DECREMENT.
: JSR PC, FIND ; SCAN COMMAND LINE
: BCS MTSET ; NOTHING ELSE THERE.
: CMPB #H, (R1) ; HALT AFTER MESSAGE COUNT EXHAUSTED.
: BEQ 4$ ; YES.
: JSR PC, ERR10 ; BAD OPTION
: BR ERPRMT ; TRY AGAIN.
:
: 4$: BIS #HALT, BASE ; SET HALT FLAG.

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

997      ;
998      ;
999      ;      PREPARE TO ENTER MAIN LOOP.
1000     ;
1001     ;
1002     ;      INITIALIZE CURRENT JUMP TABLE POINTER AND COUNTER.
1003     ;      ADJUST FOR PRE-DECREMENT OF COUNT, PRE-INCREMENT OF POINTER.
1004     ;
1005     005304      MTSET:
1006     005304      012767      000726'      172656      MOV.      #MTSUB-2, MTPNT      ;SET UP POINTER TO JUMP TABLE.
1007     005312      012767      000111      172652      MOV.      *(<TESTS*MEM>+1, MTCNT      ;AND NUMBER OF JUMP TABLE ENTRIES
1008     005320      012767      000000      176376      MOV.      #0, TD$SW      ;RESET TD.
1009     ;
1010     ;      THE STOP MESSAGE SAYS 'ENTER ANY CHARACTER TO STOP TEST(S)'.
1011     ;      ACTUALLY THE CHARACTERS W, P, C, AND T ARE SPECIAL IN THAT
1012     ;      THEY ALLOW THE RUN TO CONTINUE AFTER THE PRINTING OUT OF
1013     ;      STATUS INFORMATION. SEE THE ROUTINE 'AST'.
1014     ;
1015     005326      004767      003536      JSR.      PC, BEGTST      ;PUT OUT START TEST MESSAGE.
1016     005332      004767      003526      JSR.      PC, STOP      ;GIVE DIRECTIONS FOR STOPPING. TEST
1017     005336      Q10$S      #10, ATA, #LUN, TT, ..., <#AST>
1018     005336      005046      CLR.      -(SP)
1019     005340      005046      CLR.      -(SP)
1020     005342      005046      CLR.      -(SP)
1021     005344      005046      CLR.      -(SP)
1022     005346      005046      CLR.      -(SP)
1023     005350      012746      007206'      MOV.      #AST, -(SP)
1024     005354      005046      CLR.      -(SP)
1025     005356      005046      CLR.      -(SP)
1026     005360      005046      CLR.      -(SP)
1027     005362      012746      000001      MOV.      #LUN, TT, -(SP)
1028     005366      012746      000000G      MOV.      #10, ATA, -(SP)
1029     005372      012746      MOV.      (PC)+, -(SP)
1030     005374      001      014      .BYTE      1, 12
1031     005376      104377      EMT.      +0<377>

```

```

1019      ;
1020      ;
1021      ;      MAIN LOOP OF MEMORY TEST FUNCTION.
1022      ;
1023      ;
1024      ;      CHECK FOR THE HALT OPTION. IF THE OPTION IS
1025      ;      ON, CHECK TO SEE WHETHER ANY ERRORS HAVE OCCURRED.
1026      ;      IF THEY HAVE (AND THE HALT COUNT IS EXHAUSTED),
1027      ;      TERMINATE THE TESTS.
1028      ;
1029      005400      MTMAIN:
1030      005400      032767      000020      172416      BIT      #HALT,BASE      ;HALT AFTER ERROR.
1031      005406      001413      BEQ      1$      ;NO.
1032      005410      032767      000040      172406      BIT      #ERROR,BASE      ;HAS AN ERROR OCCURRED.
1033      005416      001407      BEQ      1$      ;NO.
1034      005420      026727      172370      000001      CMP      ERLIM,#1      ;IS REMAINING PRINTOUT COUNT 1 OR LESS
1035      005426      003003      BGT      1$      ;NO, CONTINUE.
1036      005430      004767      003424      JSR      PC,HLTTST      ;PUT OUT HALT MESSAGE.
1037      005434      000464      BR      0$      ;AND EXIT.
1038      ;
1039      ;      FIND A NON-ZERO ENTRY IN THE 'CURRENT JUMP TABLE'.
1040      ;      (A NON-ZERO ENTRY IS THE ADDRESS OF A TEST CONTROL
1041      ;      ROUTINE). IF NO NON-ZERO ENTRIES ARE FOUND BEFORE
1042      ;      THE END OF THE TABLE IS REACHED, THEN ONE MEMORY TEST
1043      ;      CYCLE OR 'PASS' IS COMPLETE.
1044      ;
1045      005436      016701      172526      1$:      MOV      MTPNT,R1      ;POINT TO JUMP TABLE.
1046      005442      005367      172524      2$:      DEC      MTCNT      ;FIRST SUB FROM # RTNS LEFT.
1047      005446      001406      BEQ      3$      ;ALL DONE, TEST LOOP FLAG.
1048      005450      005721      TST      (R1)+      ;ADVANCE POINTER.
1049      005452      005711      TST      (R1)      ;IS THERE AN ADDRESS IN THE TABLE.
1050      005454      001772      BEQ      2$      ;NO, BUMP TO NEXT.
1051      005456      010167      172506      MOV      R1,MTPNT      ;SAVE JUMP TABLE POINTER.
1052      005462      000454      BR      JMPMT      ;AND JUMP TO ROUTINE.
1053      ;
1054      ;
1055      ;      PASS FINISHED.
1056      ;
1057      ;
1058      ;      IF THE LOOP FLAG IS NOT ON, EXIT TMT.
1059      ;      IF THE LOOP FLAG IS ON AND THE LOOP COUNT IS
1060      ;      EXHAUSTED, EXIT TMT.
1061      ;
1062      005464      032767      000001      172332      3$:      BIT      #LOOP,BASE      ;IS LOOP FLAG ON.
1063      005472      001443      BEQ      7$      ;NO, GET OUT.
1064      005474      005767      172464      TST      LOOPCT      ;IS LOOP COUNT BEING USED.
1065      005500      001403      BEQ      4$      ;NO, JUST KEEP LOOPING.
1066      005502      005367      172456      DEC      LOOPCT      ;SUB FROM LOOP COUNT.
1067      005506      001435      BEQ      7$      ;FINISHED.
1068      ;
1069      ;      IF ALL TESTS, PRINT 'END OF PASS NNNN'.
1070      ;      REINITIALIZE FOR NEXT PASS.
1071      ;
1072      005510      032767      000010      172306      4$:      BIT      #ALLTST,BASE      ;ALL TESTS.
1073      005516      001415      BEQ      6$      ;
1074      005520      012702      003306      MOV      #ENDOF,R2      ;POINT TO MESSAGE.
1075      005524      012703      000014      MOV      #ENDLN,R3      ;LENGTH OF MESSAGE

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

1076 005530 012705 001465*      MOV.  #PRINT,R5      ;POINT TO PRINT LINE
1077 005534 112225              5$:  MOVB  (R2)+,(R5)+    ;MOVE MESSAGE TO PRINT LINE
1078 005536 005303              DEC.  R3
1079 005540 001375              BNE.  5$
1080 005542 004767 002742      JSR.  PC,PASSC      ;ADD # PASSES TO PRINT LINE
1081 005546 004767 003130      JSR.  PC,CONSOL     ;PRINT MESSAGE
1082.
1083 005552 062767 000001 172402 6$:  ADD.  #1,PASS      ;COUNT NUMBER OF PASSES
1084 005560 005567 172374      ADC.  PASSH      ;CARRY TO HIGH WORD
1085 005564 012767 000726* 172376  MOV.  #MTSUB-2,MTPT   ;SET POINTER TO CURRENT JUMP TABLE
1086 005572 012767 000111 172372  MOV.  *(<TESTS*MEM)+1,MTCNT ;SET COUNT
1087 005600 000677              BR.  MTMAIN      ;AND ENTER LOOP
1088
1089
1090
1091
1092.
1093 005602 004767 003246      7$:  JSR.  PC,ENDTST     ;MEMORY TEST(S) ENDED
1094 005606              8$:  EXIT$S      ;AND LEAVE
      005606 012746      MOV.  (PC)+,-(SP)
      005610 063        .BYTE  51..1
      005612 104377      EMT.  +0<377>
1095
1096
1097
1098
1099
1100
1101
1102.
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112.005614              JMPMT:
1113 005614 012700 000110*      MOV.  *(<TESTS*MEM>,R0 ;TOTAL NUMBER OF ROUTINES
1114 005620 166700 172346      SUB.  MTCNT,R0 ;SUB # RTS REMAINING TO BE EXECUTED
1115 005624 012701 000006*      MOV.  #MEM,R1 ;PUT DIVISOR IN R1 FOR SUBRTN
1116 005630 004767 000000G*      JSR.  PC,$DIV
1117 005634 010100      MOV.  R1,R0 ;SAVE TABLE POSITION IN WORD OFFSET FORM
1118
1119
1120
1121
1122.005636 022700 000005      CMP.  #5,R0 ;QUOTIENT = Z-REL TEST NUMBER
1123 005642 001050      BNE.  5$ ;NOT TEST 6 THIS TIME
1124 005644 022701 000005      CMF.  #5,R1 ;BYTE TRANSLATOR
1125 005650 001004      BNE.  1$
1126 005652 012767 000001 172150  MOV.  #1,T6FACT ;SET UP MEMORY INCREMENT
1127 005660 000435      BR.  4$
1128 005662 022701 000004      1$:  CMP.  #4,R1 ;OUTPUT BUFFER
1129 005666 001004      BNE.  2$

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

1130 005670 012767 000002 172132 MOV #2,T6FACT ;SET-UP-MEMORY-INCREMENT
1131 005676 000426 BR 4$
1132 005700 022701 000003 2$: CMP #3,R1 ;INPUT-BUFFER
1133 005704 001007 BNE 3$
1134 005706 012767 000002 172114 MOV #2,T6FACT ;SET-UP-MEMORY-INCREMENT
1135 005714 012767 000200 176360 MOV #T$DISK,TD$CTW ;SIMULATE-DISK
1136 005722 000445 BR LASTJ ;JUMP-TO-ROUTINE
1137 005724 012767 000001 172076 3$: MOV #1,T6FACT ;SET-UP-MEMORY-INCREMENT
1138 005732 012767 020000 176360 MOV #T$SCLK,TD$CTW ;START-CLOCK
1139 005740 010102 BR R1,R2 ;COPY-REMAINDER
1140 005742 006302 ASL R2 ;SHIFT-FOR-WORD-OFFSET
1141 005744 016267 000174 176360 MOV SETLBL(R2),TD$CTW ;SELECT-WHICH-FSA
1142 005752 000431 BR LASTJ ;JUMP-TO-RTN
1143 005754 012767 000040 176360 4$: MOV #T$ICD,TD$CTW ;SET-DIAGNOSTIC-BIT
1144 005762 000425 BR LASTJ ;JUMP-TO-ROUTINE
1145 ;
1146 005764 022701 000003 5$: CMP #3,R1 ;INPUT-BUFFER
1147 005770 001004 BNE 6$
1148 005772 012767 000200 176360 MOV #T$DISK,TD$CTW ;SIMULATE-DISK
1149 006000 000416 BR LASTJ
1150 006002 022701 000002 6$: CMP #2,R1 ;2-IS-MAX-OFFSET-FOR-FSA'S
1151 006005 002410 BLT 7$ ;NOT-AN-FSA
1152 006010 012767 020000 176360 MOV #T$SCLK,TD$CTW ;START-CLOCK
1153 006016 006301 ASL R1 ;CONVERT-TO-WORD-OFFSET
1154 006020 016167 000174 176360 MOV SETLBL(R1),TD$CTW ;SELECT-WHICH-FSA
1155 006026 000404 BR LASTJ1
1156 006030 012767 000040 176360 7$: MOV #T$ICD,TD$CTW ;SELECT-DIAGNOSTIC-BIT
1157 ;
1158 ;
1159 ;
1160 ;
1161 ;
1162 ;
1163 ;
1164 ;
1165 006036 006301 LASTJ: ASL R1 ;SHIFT-FOR-WORD-OFFSET
1166 006040 010100 LASTJ1: MOV R1,R0 ;LOAD-R0
1167 006042 006301 ASL R1 ;SHIFT-FOR-DBL-WORD-OFFSET
1168 006044 016146 001150 MOV CURLIM(R1),-(SP) ;MOVE-MEMORY-END-ADDR-TO-STACK
1169 006050 016146 001152 MOV CURLIM+2(R1),-(SP) ;MOVE-MEMORY-START-ADDRESS
1170 ;
1171 006054 016701 172110 MOV MTPNT,R1 ;LOAD-POINTER-TO-CONTROL-RTN-ADDR
1172 006060 000171 000000 JMP @R1 ;JUMP-TO-TEST-CONTROL-ROUTINE

```

```
1174      ;
1175      ;
1176      ;      MEMORY TEST CONTROL ROUTINES.
1177      ;
1178      ;      TESTS ARE DESCRIBED FULLY IN THE SUB-MODULE TTEST.
1179      ;
1180      ;
1181      ;      TEST-01
1182      ;
1183      ;
1184      006064      T1:
1185      006064      004770      001214'      JSR      PC,@T1ADDR(R0)
1186      006070      000167      000434      JMP      MTJUMP      ;AND RETURN TO LOOP.
1187      ;
1188      ;
1189      ;      TEST-02
1190      ;
1191      ;
1192      006074      T2:
1193      006074      005067      173174      CLR      CKDATA      ;SET TEST PATTERN TO ZERO
1194      006100      004770      001200'      JSR      PC,@STADDR(R0)
1195      006104      000167      000420      JMP      MTJUMP
1196      ;
1197      ;
1198      ;      TEST-03
1199      ;
1200      ;
1201      006110      T3:
1202      006110      012767      177777'      173156      MOV      #-1,CKDATA
1203      006116      004770      001200'      JSR      PC,@STADDR(R0)
1204      006122      000167      000402      JMP      MTJUMP
1205      ;
1206      ;
1207      ;      TEST-04
1208      ;
1209      ;
1210      006126      T4:
1211      006126      012767      125252'      173140      MOV      #125252,CKDATA
1212      006134      004770      001200'      JSR      PC,@STADDR(R0)
1213      006140      000167      000364      JMP      MTJUMP
1214      ;
1215      ;
1216      ;      TEST-05
1217      ;
1218      ;
1219      006144      T5:
1220      006144      012767      146314'      173122      MOV      #146314,CKDATA
1221      006152      004770      001200'      JSR      PC,@STADDR(R0)
1222      006156      012767      031463'      173110      MOV      #031463,CKDATA
1223      006164      004770      001200'      JSR      PC,@STADDR(R0)
1224      006170      000167      000334      JMP      MTJUMP
1225      ;
1226      ;
1227      ;      TEST-06
1228      ;
1229      ;
1230      006174      T6:
```



```

1231      :      CLR      CKDATA      :SET PATTERN TO ZERO
1232      :      JSR      PC,@STADDR(R0)
1233      :      JSR      PC,@T6ADDR(R0)
1234      :      ADD      T6FACT,(SP)      :PERFORM AT NEXT ADDRESS
1235      :      CLR      CKDATA      :RESET PATTERN TO ZERO
1236      :      JSR      PC,@STADDR(R0)
1237      :      JSR      PC,@T6ADDR(R0)
1238      :      SUB      T6FACT,(SP)      :RESTORE ORIGINAL LOWER LIMIT
1239 006174 000167 000330      JMP      MTJUMP      :RETURN TO TOP OF MT COMMAND LOOP
1240      :
1241      :
1242      :      TEST 07
1243      :
1244      :
1245 006200      T7:
1246 006200 004770 001230      JSR      PC,@T7ADDR(R0)
1247 006204 000167 000320      JMP      MTJUMP      :RETURN TO TOP OF MT COMMAND LOOP
1248      :
1249      :
1250      :      TEST 08
1251      :
1252      :
1253 006210      T8:
1254 006210 012767 000377 173056      MOV      #377,CKDATA      :SET TEST PATTERN = X'00FF'
1255 006216 004770 001200      JSR      PC,@STADDR(R0)
1256 006222 012767 177400 173044      MOV      #177400,CKDATA      :SET TEST PATTERN = X'FF00'
1257 006230 004770 001200      JSR      PC,@STADDR(R0)
1258 006234 000167 000270      JMP      MTJUMP      :RETURN TO TOP OF MT COMMAND LOOP
1259      :
1260      :
1261      :      TEST 09
1262      :
1263      :
1264 006240      T9:
1265 006240 012767 000001 173026      MOV      #1,CKDATA      :START TEST PATTERN AT 1
1266 006246 004770 001200      JSR      PC,@STADDR(R0)
1267 006252 016702 173016      MOV      CKDATA,R2      :LOAD FOR SHIFT
1268 006256 006302      ASL      R2      :SHIFT A BIT
1269 006260 010267 173010      MOV      R2,CKDATA      :NEXT TEST PATTERN
1270 006264 005702      TST      R2      :FINISHED?
1271 006266 001367      BNE      1$
1272 006270 000167 000234      JMP      MTJUMP      :RETURN TO LOOP
1273      :
1274      :
1275      :      TEST 10
1276      :
1277      :
1278 006274      T10:
1279      :
1280      :      MOV      MT10,CKDATA      :USER PATTERN
1281 006274 000167 000230      JSR      PC,@STADDR(R0)
1282      :      JMP      MTJUMP      :RETURN TO TOP OF MT COMMAND LOOP
1283      :
1284      :
1285      :      TEST 11
1286      :
1287 006300      T11:

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

1288 006300 012767 100001 172766      MOV.    #100001,CKDATA
1289 006306 004770 001200      JSR.    PC,@STADDR(R0)
1290 006312 012767 040002 172754      MOV.    #040002,CKDATA
1291 006320 004770 001200      JSR.    PC,@STADDR(R0)
1292 006324 012767 020004 172742      MOV.    #020004,CKDATA
1293 006332 004770 001200      JSR.    PC,@STADDR(R0)
1294 006336 012767 001010 172730      MOV.    #001010,CKDATA
1295 006344 004770 001200      JSR.    PC,@STADDR(R0)
1296 006350 012767 004020 172716      MOV.    #004020,CKDATA
1297 006356 004770 001200      JSR.    PC,@STADDR(R0)
1298 006362 012767 002040 172704      MOV.    #002040,CKDATA
1299 006370 004770 001200      JSR.    PC,@STADDR(R0)
1300 006374 012767 001100 172672      MOV.    #001100,CKDATA
1301 006402 004770 001200      JSR.    PC,@STADDR(R0)
1302 006406 012767 000600 172660      MOV.    #000600,CKDATA
1303 006414 004770 001200      JSR.    PC,@STADDR(R0)
1304 006420 000167 000104      JMP.    MTJUMP
1305
1306
1307
1308
1309
1310 006424
1311 006424 005067 172644      T12:
1312 006430 004770 001200      CLR.    CKDATA
1313 006434 005067 172636      JSR.    PC,@STADDR(R0)
1314 006440 012767 177777 172632      CLR.    CK2
1315 006446 004770 001244      MOV.    #-1,CK3
1316 006452 012767 177777 172616      JSR.    PC,@TCUADD(R0)
1317 006460 005067 172614      MOV.    #-1,CK2
1318 006464 004770 001244      CLR.    CK3
1319
1320 006470 005067 172602      JSR.    PC,@TCUADD(R0)
1321 006474 012767 177777 172576      CLR.    CK2
1322 006502 004770 001200      MOV.    #-1,CK3
1323 006506 012767 177777 172562      JSR.    PC,@TCUADD(R0)
1324 006514 005067 172560      MOV.    #-1,CK2
1325 006520 004770 001200      CLR.    CK3
1326 006524 000167 000000      JSR.    PC,@TCUADD(R0)
1327
1328
1329 006530 062706 000004      JMP.    MTJUMP
1330 006534 000167 176540      ADD.    #4,SP
                                JMP.    MTMAIN
                                ;RESTORE SP.
                                ;AND- RETURN.

```

```

1332.
1333.
1334.
1335.
1336.
1337 006540
1338 006540
      006540 010046
      006542 010146
      006544 010246
      006546 010346
      006550 010446
      006552 010546
1339.
1340.
1341.
1342.
1343.
1344.
1345 006554 052767 000040 171242.
1346 006562 005767 171226
1347 006566 001412.
1348 006570 003002.
1349 006572 000167 000372
1350.
1351.
1352.
1353.
1354.
1355 006576 005367 171212
1356 006602 001004
1357 006604 005367 171204
1358 006610 000167 000354
1359.
1360.
1361.
1362.
1363.
1364.
1365.
1366.
1367.
1368.
1369 006614 012700 000110
1370 006620 166700 171346.
1371 006624 012701 000006.
1372 006630 004767 000000G.
1373 006634 010146
1374 006636 010001
1375 006640 005201
1376 006642 012700 001316'
1377 006646 012702 000001
1378 006652 004767 000000G.
1379.
1380 006656 012705 001465'
1381 006662 012700 001323'
1382 006666 012701 000005

```

MEMORY TEST ERROR ROUTINE

MEMERR::

```

      SAVE R0,R1,R2,R3,R4,R5
      MOV R0,-(SP)
      MOV R1,-(SP)
      MOV R2,-(SP)
      MOV R3,-(SP)
      MOV R4,-(SP)
      MOV R5,-(SP)

      SET FLAG FOR ERROR ENCOUNTERED. AN ERROR MESSAGE LIMIT
      COUNT OF ZERO MEANS THAT THE COUNT IS NOT BEING USED.
      A COUNT OF -1 MEANS THAT THE LIMIT HAS BEEN REACHED.
      (NO MORE ERROR MESSAGES ARE TO BE PRINTED).

      BIS #ERROR.BASE ;SET FLAG FOR ERROR ENCOUNTERED
      TST ERLIM
      BEQ 2$
      BGT 1$
      JMP ERRORX

      DECREMENT ERROR LIMIT COUNT. IF IT GOES ZERO HERE, SET
      NO MORE ERROR MESSAGES.
      IT TO -1.

      1$: DEC ERLIM
      BNE 2$
      DEC ERLIM
      JMP ERRORX

      DERIVE TEST NUMBER FROM THE ADDRESS POSITION OF THE CURRENT
      MEMORY TEST CONTROL ROUTINE IN THE CURRENT JUMP TABLE.

      GET THE OFFSET FROM THE BEGINNING OF THE TABLE. DIVIDE
      BY THE NUMBER OF MEMORIES IN THE TABLE. THE QUOTIENT
      IS THE TEST NUMBER (ZERO-RELATIVE). THE REMAINDER IS THE
      MEMORY PLACE-HOLDER VALUE. SAVE IT.

      2$: MOV #<TESTS*MEM>,R0 ;LOAD NUMBER OF TESTS (TOTAL)
      SUB MTCT,R0 ;SUB CURRENT COUNT OF TABLE ENTRIES
      MOV #MEM,R1 ;SYSTEM SUBRTN EXPECTS DIVISOR IN R1
      JSR PC,$DIV ;DIVIDEND IS IN R1
      MOV R1,-(SP) ;TEMP SAVE REMAINDER
      MOV R0,R1 ;PREPARE FOR CONVERSION TO DECIMAL
      INC R1 ;ADJUST ZERO-RELATIVE NUMBER
      MOV #ASWRK,R0 ;CONVERT INTO WORK FIELD
      MOV #1,R2 ;SUPPLY LEADING ZEROS
      JSR PC,$CBDSG ;CONVERT BINARY TO DECIMAL

      MOV #PRINT,R5 ;POINT AT PRINT LINE
      MOV #TMSG,R0 ;POINT AT TEST
      MOV #5,R1 ;NUMBER OF BYTES IN STRING

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

1383 006672 112025      3$:  MOV.  (R0)+,(R5)+      ;MOVE LABEL TO PRINT LINE
1384 006674 005301      DEC.  R1
1385 006676 001375      BNE.  3$
1386
1387 006700 116725 172415      MOV.  ASWRK+3,(R5)+      ;MOVE A DIGIT OF TEST NUMBER
1388 006704 116725 172412      MOV.  ASWRK+4,(R5)+
1389 006710 062705 000002      ADD.  #2,R5              ;BUMP PRINT LINE POINTER
1390
1391      ;
1392      ;
1393 006714 012700 001330      MOV.  #PMMSG,R0          ;POINT AT "PASS"
1394 006720 012701 000005      MOV.  #5,R1            ;NUMBER OF BYTES IN STRING
1395 006724 112025      4$:  MOV.  (R0)+,(R5)+      ;MOVE LABEL TO PRINT LINE
1396 006726 005301      DEC.  R1
1397 006730 001375      BNE.  4$
1398 006732 004767 001552      JSR.  PC,PASSC          ;CONVERT NUMBER OF PASSES
1399 006736 062705 000003      ADD.  #3,R5              ;BUMP POINTER
1400
1401      ;
1402      ;
1403 006742 012700 001335      MOV.  #FMSG,R0          ;POINT AT STRING
1404 006746 012701 000020      MOV.  #16,R1          ;LENGTH OF STRING
1405 006752 112025      5$:  MOV.  (R0)+,(R5)+      ;MOVE LABEL TO PRINT LINE
1406 006754 005301      DEC.  R1
1407 006756 001375      BNE.  5$
1408
1409      ;
1410      ;
1411      ;
1412 006760 012601      MOV.  (SP)+,R1          ;LOAD REMAINDER FROM DIVISION ABOVE
1413 006762 012700      MOV.  #6,R0            ;SYSTEM SUBRTN EXPECTS MULTIPLIER IN R0
1414 006766 004767 000000G      JSR.  PC,MUL
1415 006772 012700 001417      MOV.  #FTBL,R0          ;POINT TO TABLE OF MEMORY NAMES
1416 006776 060100      ADD.  R1,R0            ;POINT TO THE ONE THAT FAILED
1417 007000 012701 000006      MOV.  #6,R1            ;NUMBER OF CHARS IN NAME
1418 007004 112025      6$:  MOV.  (R0)+,(R5)+      ;MOVE NAME TO PRINT LINE
1419 007006 005301      DEC.  R1
1420 007010 001375      BNE.  6$
1421 007012 004767 001664      JSR.  PC,CONSOL        ;WRITE ONE LINE TO CONSOLE
1422
1423      ;
1424      ;
1425      ;
1426 007016 012705 001465      MOV.  #PRINT,R5        ;POINT TO PRINT LINE
1427 007022 012700 001355      MOV.  #MSG,R0          ;POINT TO ADDRESS
1428 007026 012701 000011      MOV.  #9,R1            ;LOAD LENGTH OF STRING
1429 007032 112025      7$:  MOV.  (R0)+,(R5)+      ;MOVE LABEL
1430 007034 005301      DEC.  R1
1431 007036 001375      BNE.  7$
1432
1433 007040 016701 172240      MOV.  ERRADD,R1          ;LOAD ERROR ADDRESS
1434 007044 004767 001404      JSR.  PC,UNPK          ;CONVERT TO PRINTABLE CHARS
1435 007050 005205      INC.  R5              ;BUMP PRINT LINE POINTER
1436
1437 007052 012700 001366      MOV.  #EMSG,R0          ;POINT TO EXPECTED
1438 007056 012701 000012      MOV.  #10,R1           ;LOAD NUMBER OF CHARS
1439 007062 112025      8$:  MOV.  (R0)+,(R5)+      ;MOVE LABEL

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

1440 007064 005301          DEC----- R1
1441 007066 001375          BNE      8$
1442
1443 007070 016701 172200:    MOV      CKDATA,R1      ;LOAD MEMORY TEST PATTERN
1444 007074 004767 001354:    JSR      PC,UNPK      ;CONVERT
1445 007100 005205          INC      R5      ;BUMP PRINT LINE POINTER
1446
1447 007102 012700 001400*    MOV      #RMSG,R0      ;POINT TO RECEIVED
1448 007106 012701 000012:    MOV      #10,R1      ;LOAD NUMBER OF CHARS
1449 007112 112025          9$:    MOV     (R0)+,(R5)+    ;MOVE LABEL
1450 007114 005301          DEC      R1
1451 007116 001375          BNE      9$
1452
1453
1454          ;    COUNT OF NUMBER OF WORDS TO PRINT = 0 SIGNALS THAT
1455          ;    THE RESULTS OF SEVERAL MEMORY READS WERE UNPREDICTABLE
1456          ;    PRINT * * *
1457 007120 016700 172162:    MOV      ERCT,R0      ;LOAD NUMBER OF WORDS TO CONVERT
1458 007124 003010          BGT      11$      ;PRINT WORDS
1459 007126 012700 001412*    MOV      #UNMSG,R0    ;* * *
1460 007132 012701 000005:    MOV      #5,R1      ;LENGTH OF MESSAGE
1461 007136 112025          10$:   MOV     (R0)+,(R5)+
1462 007140 005301          DEC      R1
1463 007142 001375          BNE      10$
1464 007144 000407          BR      13$
1465
1466 007146 012702 001310*    11$:   MOV     #ERW1,R2    ;POINT TO FIRST OF THEM
1467 007152 012201          12$:   MOV     (R2)+,R1    ;LOAD THE WORD ITSELF
1468 007154 004767 001274          JSR      PC,UNPK
1469 007160 005300          DEC      R0
1470 007162 001373          BNE      12$
1471
1472 007164 004767 001512          13$:   JSR      PC,CONSOL    ;ELSE WRITE TO CONSOLE
1473
1474 007170          ;
1475 007170          ;    ERRORX:
1476 007170 012605          RESTOR  R0,R1,R2,R3,R4,R5
1477 007172 012604          MOV      (SP)+,R5
1478 007174 012603          MOV      (SP)+,R4
1479 007176 012602          MOV      (SP)+,R3
1480 007200 012601          MOV      (SP)+,R2
1481 007202 012600          MOV      (SP)+,R1
1482 007204 000207          MOV      (SP)+,R0
1483          RTS      PC

```

Approved For Release 2005/07/11 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/10 : CIA-RDP85-00514R000200020001-3


```
1643 007640 001375 BNE.... 2$
1644
1645 007642 004767 000642 JSR PC,PASSC ;CONVERT DOUBLE WORD
1646 007646 004767 001030 JSR PC,CONSOL
1647
1648 REST: RESTOR R0,R1,R2,R3,R4,R5
      MOV (SP)+,R5
      MOV (SP)+,R4
      MOV (SP)+,R3
      MOV (SP)+,R2
      MOV (SP)+,R1
      MOV (SP)+,R0
1649
1650 007666 122767 000120 170110 CMPB #P,ASTURD ;PRINT # PASSES AND CONTINUE
1651 007674 001414 BEQ 10$
1652 007676 122767 000103 170100 CMPB #C,ASTURD ;PRINT TEST PATTERN AND CONTINUE
1653 007704 001410 BEQ 10$
1654 007706 122767 000124 170070 CMPB #T,ASTURD ;PRINT TEST NUMBER AND CONTINUE
1655 007714 001404 BEQ 10$
1656 007716 122767 000127 170060 CMPB #W,ASTURD ;PRINT WHERE AND CONTINUE
1657 007724 001003 BNE 1$ ;NO EXIT
1658
1659 007726 10$: ASTX$S
      MOV (PC)+,-(SP)
      BYTE 115,,1
      EMT +0<377>
1660
1661 007734 012701 001465 1$: MOV #PRINT,R1 ;POINT TO PRINT LINE
1662 007740 012700 000116 MOV #78,,R0 ;NUMBER OF BYTES
1663 007744 112711 000040 2$: MOVB #40,(R1) ;CLEAR LINE TO BLANKS
1664 007750 005300 R0
1665 007752 001374 BNE 2$
1666 007754 112767 000015 171503 MOVB #15,PRINT ;WRITE OUT 1 CR+LF
1667 007762 112767 000012 171476 MOVB #12,PRINT+1
1668 007770 004767 000706 JSR PC,CONSOL
1669
1670 007774 EXIT$S
      MOV (PC)+,-(SP)
      BYTE 51,,1
      EMT +0<377>
      007776 063 001
      010000 104377
```

TMT MACRO M1110 27-MAR-80 15:38 PAGE 22

```

1672.      :
1673.      :
1674.      :      SUBRTN FOR MEMORY LIMITS.
1675.      :
1676.      :      INPUT:
1677.      :      2(SP)  ABSOLUTE LOWER LIMITS.
1678.      :      4(SP)  ABSOLUTE UPPER LIMITS.
1679.      :      6(SP)  ADDRESS OF PROMPT ROUTINE.
1680.      :
1681.      :      OUTPUT:
1682.      :      (SP)   CURRENT WORKING LOWER LIMITS.
1683.      :      2(SP)  CURRENT WORKING UPPER LIMITS.
1684.      :
1685.      :      WORK FIELDS USED:
1686.      :      LOWER.
1687.      :      UPPER.
1688.      :
1689.      :      PROMPT FOR LIMITS. A <CR> RESPONSE MEANS TO TAKE ALL
1690.      :      THE ABSOLUTE LOWER AND UPPER LIMITS AND RETURN
1691.      :      THEM ON THE STACK. OTHERWISE IF THE RESPONSE IS IN
1692.      :      THE FORM:
1693.      :
1694.      :      >000000 000012.
1695.      :
1696.      :      THIS ROUTINE CONVERTS THE FIRST VALUE AND COMPARES IT
1697.      :      AGAINST THE MEMORY'S ABSOLUTE LOWER LIMITS AT 2(SP). IF
1698.      :      THE NEW LIMITS ARE IN RANGE, THEY ARE PLACED IN A
1699.      :      TEMPORARY WORK FIELD. THE ROUTINE THEN CHECKS THE COMMAND
1700.      :      LINE FOR THE UPPER LIMITS, CONVERTS THEM, AND COMPARES
1701.      :      THEM AGAINST THE MEMORY'S ABSOLUTE UPPER LIMITS. IF THE
1702.      :      NEW LIMITS ARE IN RANGE, THEY ARE PLACED IN A TEMPORARY
1703.      :      WORK AREA. THE ROUTINE THEN COMPARES THE NEW UPPER LIMITS
1704.      :      WITH THE NEW LOWER LIMITS. IF THE NEW UPPER LIMITS ARE
1705.      :      EQUAL TO OR GREATER THAN THE NEW LOWER LIMITS, BOTH NEW
1706.      :      VALUES ARE PLACED ON THE STACK. THE ROUTINE THAT CALLED
1707.      :      LIMITS WILL TAKE THESE VALUES OFF THE STACK AND PLACE
1708.      :      THEM IN THE 'CURRENT LIMITS TABLE'. DURING THE MEMORY
1709.      :      TEST CYCLE, THE LIMITS FROM THIS TABLE ARE MADE AVAILABLE
1710.      :      TO THE MEMORY TEST ROUTINES.
1711.      :
1712.      :
1713.      :      LIMITS:
1714.      :      010002.      JSR.      PC,06(SP)      ;PROMPT FOR LIMITS
1715.      :      004776      JSR.      PC,FIND.      ;FIRST FIND A NUMBER.
1716.      :      004767      BCC.      LIMX2.      ;NO OVERRIDES, LEAVE LIMITS ALONE.
1717.      :      000372      JSR.      PC,PACK.      ;CONVERT LOWER LIMIT TO BINARY.
1718.      :      103003      BCC.      1$.      ;VALUE OK, CONTINUE.
1719.      :      004767      JSR.      PC,ERR4
1720.      :      000765      BR.      LIMITS.      ;TRY AGAIN.
1721.      :
1722.      :      CHECK LOWER LIMITS.
1723.      :
1724.      :      010030      026766      167762      000002.  1$.  CMP.      BINWD,2(SP)      ;COMPARE LOWER LIMITS.
1725.      :      103003      BHIS.      2$.      ;OK, CONTINUE.
1726.      :      004767      JSR.      PC,ERR4
1727.      :      000756      BR.      LIMITS.      ;TRY AGAIN.
1728.      :

```

```

1729      ;      ALSO CHECK NEW LOWER LIMITS AGAINST ABSOLUTE UPPER LIMITS.
1730      ;      ON THE STACK. AN ERROR HERE WOULD SHOW UP BELOW BUT IT
1731      ;      IS MORE CORRECT TO REPORT AN ERROR IN LOWER LIMITS IF
1732      ;      IF THE NEW LOWER LIMITS ARE HIGHER THAN THE ABSOLUTE
1733      ;      UPPER LIMITS.
1734      ;
1735 010046 026766 167744 000004 2$: CMP BINWD,4(SP) ;IS NEW LOW LIMIT GT ABS UPPER LIMIT
1736 010054 101403 BLOS 20$ ;NO, NEW LOWER LIMIT IS OK
1737 010056 004767 001056 JSR PC,ERR4
1738 010062 000747 BR LIMITS
1739      ;
1740      ;      SAVE NEW LOWER LIMITS. FIND NEW UPPER LIMITS IN COMMAND
1741      ;      LINE.
1742      ;
1743 010064 016767 167726 167730 20$: MOV BINWD,LOWER ;MOVE IN NEW LOW LIMITS
1744 010072 004767 000176 JSR PC,FIND ;FIND UPPER LIMITS IN COMMAND LINE
1745 010076 103003 BCC 3$ ;OK, CONTINUE
1746 010100 004767 001030 JSR PC,ERR5
1747 010104 000736 BR LIMITS ;START OVER
1748      ;
1749 010106 004767 000300 3$: JSR PC,PACK ;CONVERT UPPER LIMITS
1750 010112 103003 BCC 4$ ;OK, CONTINUE
1751 010114 004767 001014 JSR PC,ERR5
1752 010120 000730 BR LIMITS ;START OVER
1753      ;
1754      ;      CHECK NEW UPPER LIMITS.
1755      ;
1756 010122 026766 167670 000004 4$: CMP BINWD,4(SP) ;COMPARE UPPER LIMITS
1757 010130 101403 BLOS 5$ ;OK, CONTINUE
1758 010132 004767 000776 JSR PC,ERR5
1759 010136 000721 BR LIMITS ;START OVER
1760      ;
1761      ;      COMPARE NEW LOWER LIMITS WITH NEW UPPER LIMITS.
1762      ;
1763 010140 016767 167652 167652 5$: MOV BINWD,UPPER ;PLACE NEW UPPER LIMIT ON STACK
1764 010146 026767 167650 167644 CMP LOWER,UPPER ;IS UPPER LIMIT GT LOWER LIMIT
1765 010154 101403 BLOS LIMX ;YES, EXIT
1766 010156 004767 000752 JSR PC,ERR5
1767 010162 000707 BR LIMITS ;TRY AGAIN
1768      ;
1769 010164 016766 167632 000002 LIMX: MOV LOWER,2(SP) ;PUT NEW LOWER LIMITS ON STACK
1770 010172 016766 167622 000004 MOV UPPER,4(SP) ;PUT NEW UPPER LIMITS ON STACK
1771 010200 000207 LIMX2: RTS PC

```

```

1773      ;
1774      ;
1775      ;
1776      ;
1777      ;
1778      ;
1779      ;
1780      ;
1781      ;
1782      ;
1783      ;
1784      ;
1785      ;
1786      ;
1787      ;
1788      ;
1789      ;
1790      ;
1791      ;
1792      ;
1793      ;
1794      ;
1795      ;
1796      ;
1797      ;
1798      ;
1799      ;
1800      ;
1801      ;
1802      ;
1803      ;
1804      ;
1805      ;
1806      ;
1807      ;
1808      ;
1809      ;
1810      ;
1811      ;
1812      ;
1813      ;
1814      ;
1815      ;
1816      ;
1817      ;

SCAN: A TABLE FOR A VALID COMMAND/MNEMONIC.

INPUT:
R0 = NUMBER OF ENTRIES IN COMMAND TABLE.
R1 -> CHAR STRING IN GCML COMMAND LINE.
R2 -> TOP OF COMMAND TABLE.

OUTPUT:
R1 -> ROUTINE THAT GOVERNS THE COMMAND (IF MATCH WAS MADE).
R1 -> CHAR STRING IN COMMAND LINE (IF NO MATCH WAS MADE).
R0 = RELATIVE POSITION OF MATCHED ENTRY IN TABLE.

SCAN:
MOV. R3, -(SP)      ;SAVE R3
MOV. R0, -(SP)      ;SAVE # ENTRIES
MOV. R1, -(SP)      ;SAVE POINTER TO BEGINNING OF STRING.

FNOUT1: MOV. (SP), R1      ;POINT TO NON-BLANK IN COMMAND LINE.
MOV. #2, R3          ;NUMBER OF CHARS IN NON-BLANK FIELD.
FNIN1:  CMPB. (R1)+, (R2)+ ;DOES COMMAND LINE MATCH TABLE ENTRY.
BNE. FNOUT2          ;NO, TRY NEXT TABLE ENTRY.
DEC. R3              ;SUB FROM LOOP COUNT.
BNE. FNIN1
BR. FNMTCH          ;COMMAND FOUND IN TABLE.
FNOUT2: ADD. R3, R2      ;ADD # UNCOMPARED CHARS TO POINTER.
INC. R2              ;THEN ADJUST TO NEXT TABLE ENTRY.
DEC. R0              ;SUB FROM OUTER LOOP COUNT.
BNE. FNOUT1          ;TRY AGAIN.
MOV. (SP)+, R1        ;RESTORE COMMAND LINE POINTER.
MOV. (SP)+, R0        ;RESTORE R0
MOV. (SP)+, R3        ;RESTORE R3
SEC.                 ;COMMAND NOT IN TABLE.
RTS. PC.

FNMTCH: MOV. R2, R1      ;POINT R1 AT MEMORY FLAG IN TABLE.
ADD. #2, SP          ;POINT TO INCOMING R0 ON STACK.
MOV. (SP)+, R2        ;GET TOTAL # TABLE ENTRIES.
SUB. R0, R2           ;GET POSITION OF MATCHED ENTRY.
MOV. R2, R0           ;PUT IN R0 FOR RETURN.
MOV. (SP)+, R3        ;RESTORE R3
CLC.
RTS. PC.

```

```

1819      ;
1820      ;
1821      ;
1822      ;
1823      ;
1824      ;
1825      ;
1826      ;
1827      ;
1828      ;
1829      ;
1830      ;
1831      ;
1832      ;
1833      ;
1834      ;
1835      ;
1836      ;
1837      ;
1838      ;
1839      ;
1840      ;
1841      ;
1842      ;
1843      ;
1844      ;
1845      ;
1846      ;
1847      ;
1848      ;
1849      ;
1850      ;
1851      ;
1852      ;
1853      ;
1854      ;
1855      ;
1856      ;
1857      ;
1858      ;
1859      ;
1860      ;
1861      ;
1862      ;
1863      ;
1864      ;
1865      ;
1866      ;
1867      ;
1868      ;
1869      ;
1870      ;
1871      ;
1872      ;

```

FIND THE NEXT NON-BLANK IN THE COMMAND BUFFER.
 THEN FIND THE LENGTH OF THE STRING THAT STARTS WITH THAT CHARACTER.

INPUT:
 GCMLN - NUMBER OF UNPROCESSED BYTES IN COMMAND LINE.
 GCMPT - ADDR OF NEXT UNPROCESSED POSITION IN COMMAND LINE.

OUTPUT:
 R1 -> STRING. R0 = LENGTH OF STRING.
 GCMLN, GCMPT UPDATED FOR NEXT ENTRY INTO THIS ROUTINE.

THIS ROUTINE IS DESIGNED TO BE ENTERED A NUMBER OF TIMES
 IN THE PARSING OF A COMMAND LINE. THE FIELDS GCMLN AND
 GCMPT ARE REFRESHED WHEN A NEW COMMAND LINE IS READ
 (SEE THE MESSAGE PRINTING/PROMPTING CODE).

```

FIND:
MOV.    R2, -(SP)          ;SAVE R2
MOV.    GCMLN, R1          ;#. BYTES REMAINING IN COMMAND BUFFER.
BEQ.    FSECK              ;THERE ARE NONE.
MOV.    GCMPT, R2          ;LOAD CURRENT POINTER.
1$:     CMPB. #40, (R2)     ;LOOK FOR A BLANK.
        BEQ.    10$        ;OK. BUMP TO NEXT CHAR.
        CMPB. #',, (R2)    ;COMMA IN COMMAND LINE.
        BNE.    2$        ;TREAT COMMA AS BLANK.
        INC.    R2         ;BUMP POINTER.
        DEC.    R1         ;SUB FROM REMAINING LENGTH.
        BNE.    1$
        BR      FSECK      ;NO NON-BLANK FOUND.

2$:     MOV.    R2, -(SP)   ;TEMP SAFE POINTER TO BEGINNING OF STRING
        CLR.    R0         ;CLEAR CHAR COUNT.
3$:     CMPB. #40, (R2)    ;LOOK FOR A BLANK.
        BEQ.    4$        ;FOUND END OF STRING.
        CMPB. #',, (R2)    ;TREAT COMMAS AS BLANKS.
        BEQ.    4$
        INC.    R2         ;BUMP POINTER.
        INC.    R0         ;BUMP CHAR COUNT.
        DEC.    R1         ;SUB FROM BYTES REMAINING.
        BNE.    3$

4$:     MOV.    R2, GCMPT   ;SAVE POINTER FOR NEXT TIME.
        MOV.    R1, GCMLN   ;SAVE BYTES REMAINING FOR NEXT TIME.
        MOV.    (SP)+, R1   ;POINTER TO BEGINNING OF STRING.
        MOV.    (SP)+, R2   ;RESTORE R2.
        CLC.
        RTS.    PC

FSECK:  MOV.    (SP)+, R2   ;RESTORE R2.
        SEC.
        RTS.    PC

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

1874      ;
1875      ;
1876      ;
1877      ;   CONVERT AN ASCII OCTAL VALUE FROM THE COMMAND LINE INTO BINARY
1878      ;   LEGAL STRINGS CONTAIN FROM 1 TO 6 CHARACTERS.
1879      ;
1880      ;   INPUT:
1881      ;   R0 = NUMBER OF CHARACTERS IN ASCII STRING
1882      ;   R1 -> STRING
1883      ;
1884      ;   OUTPUT:
1885      ;   THE FIELD 'BINWD' CONTAINS THE CONVERTED VALUE.
1886      ;
1887 010412      ;
1888 010412 022700 000006      ;   PACK:
1889 010416 001014      ;   CMP. #6,R0      ;UPPER LIMIT ON OCTAL DIGITS
1890 010420 010146      ;   BNE. PSECK.    ;ERROR EXIT
1891 010422 060016      ;   MOV. R1, -(SP) ;SAVE STRING POINTER TEMPORARILY
1892 010424 010100      ;   ADD. R0, (SP)  ;ADD CHAR COUNT
1893 010426 004767 000000G    ;   MOV. R1,R0     ;GET STRING ADDR INTO R0 FOR SUBRTN
1894 010432 005300      ;   JSR. PC,$COTB  ;CONVERT ASCII OCTAL
1895 010434 020026      ;   DEC. R0       ;SUBRTN PUSHES R0 1 TOO FAR
1896 010436 001004      ;   CMP. R0, (SP)+ ;FULL STRING CONVERTED
1897 010440 010167 167352    ;   BNE. PSECK.    ;NO ERROR
1898      ;   MOV. R1,BINWD ;SAVE CONVERTED VALUE
1899 010444 000241      ;
1900 010446 000207      ;   PCLCX: CLC.
1901 010450 000261      ;   RTS. PC.
1902 010452 000207      ;   PSECK: SEC.
1903      ;   RTS. PC.
1904      ;
1905      ;
1906      ;   CONVERT A VALUE FROM BINARY TO PRINTABLE FORM.
1907      ;   R1 = WORD TO BE CONVERTED
1908      ;   R5 -> PRINT LINE
1909      ;
1910      ;
1911 010454      ;   UNPK:
1912 010454      ;   SAVE. R0,R1,R2
1913      ;   MOV. R0, -(SP)
1914      ;   MOV. R1, -(SP)
1915      ;   MOV. R2, -(SP)
1916 010462 010500      ;
1917 010464 012702 000001      ;   MOV. R5,R0     ;PREPARE TO CALL SYSTEM SUBRTN
1918 010470 004767 000000G    ;   MOV. #1,R2     ;KEEP LEADING ZEROS
1919 010474 010005      ;   JSR. PC,$CBOMG  ;CONVERT TO OCTAL ASCII
1920 010476 005205      ;   MOV. R0,R5     ;RESTORE PRINT LINE POINTER
1921      ;   INC. R5   ;AND BUMP IT
1922 010500      ;
1923 010500 012602      ;   RESTOR. R0,R1,R2
1924 010502 012601      ;   MOV. (SP)+,R2
1925 010504 012600      ;   MOV. (SP)+,R1
1926 010506 000207      ;   MOV. (SP)+,R0
1927      ;   RTS. PC

```

```

1923      ;
1924      ;
1925      ; CONVERT NUMBER OF PASSES TO ASCII DECIMAL
1926      ; INCLUDE DOUBLE WORD CONVERSION (BINARY TO ASCII DECIMAL)
1927      ;
1928      ; INPUT: R5 -> PRINT LINE
1929      ; FIELD 'PASS' = LOW ORDER PASS COUNT
1930      ; FIELD 'PASSH' = HIGH ORDER PASS COUNT
1931      ;
1932      ; OUTPUT: R5 -> NEXT PRINT LINE POSITION AFTER CONVERTED VALUE
1933      ;
1934      ;
1935      PASSC:
1936      010510      005767      167444      TST      PASSH      ; IS PASS NUMBER A DOUBLE WORD
1937      010514      001011      BNE      DOUBLE      ; YES, CONVERT DOUBLE WORD
1938      010516      010500      MOV      R5,R0      ; PREPARE TO CALL CONVERSION RTN
1939      010520      016701      167436      MOV      PASS,R1      ; NUMBER TO BE CONVERTED
1940      010524      012702      000001      MOV      #1,R2      ; SUPPLY LEADING ZEROS
1941      010530      004767      000000G      JSR      PC,$CBDSG
1942      010534      010005      MOV      R0,R5      ; RESTORE PRINT LINE POINTER
1943      010536      000453      BR      PASSX      ; EXIT
1944      ;
1945      010540      016701      167414      DOUBLE: MOV      PASSH,R1      ; LOAD HIGH WORD
1946      010544      016702      167412      MOV      PASS,R2      ; LOAD LOW WORD
1947      010550      010346      MOV      R3,-(SP)
1948      010552      010446      MOV      R4,-(SP)
1949      ;
1950      010554      012703      000012      MOV      #10,R3      ; R3 = LOOP COUNTER
1951      010560      012704      010670      MOV      #ASCNST,R4      ; R4 -> CONVERSION STORAGE AREA
1952      010564      112724      000060      1$: MOV      MOV      #60,(R4)+      ; CLEAR STORAGE AREA TO 0'S
1953      010570      005303      DEC      R3
1954      010572      001374      BNE      1$
1955      ;
1956      010574      012704      010702      MOV      #ASCNST+12,R4      ; R4 -> LAST BYTE OF STORAGE
1957      010600      012703      000012      MOV      #12,R3
1958      010604      012700      000012      2$: MOV      #10,R0
1959      010610      010610      004767      000000G      CALL      $DDIV
1960      010614      062700      000060      JSR      PC,$DDIV
1961      010620      110044      ADD      #60,R0      ; ASCII NO. (REMAINDER+60 OCTAL)
1962      010622      005303      MOV      R0,-(R4)
1963      010624      001367      DEC      R3
1964      BNE      2$
1965      ;
1966      010626      012703      000012      MOV      #12,R3
1967      010632      122724      000060      3$: CMP      #60,(R4)+      ; REMOVE LEADING ZEROS
1968      010636      001003      BNE      4$
1969      010640      005303      DEC      R3
1970      010642      001373      BNE      3$
1971      010644      005203      INC      R3      ; IF ALL 0'S, THE LAST ONE IS OK
1972      ;
1973      010646      005304      4$: DEC      R4
1974      010650      010401      MOV      R4,R1      ; R1 -> RESULT
1975      010652      010302      MOV      R3,R2      ; R2 = LENGTH OF RESULT
1976      ;
1977      010654      112125      5$: MOV      (R1)+,(R5)+      ; MOVE CONVERTED VALUE TO PRINT LINE
1978      010656      005302      DEC      R2
1979      010660      001375      BNE      5$

```

TMT-----MACRO-M1110 27-MAR-80 15:38 PAGE-26-1

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

1979			
1980	010662	012604	MOV (SP)+,R4
1981	010664	012603	MOV (SP)+,R3
1982	010666	000207	PASSX: RTS PC
1983			
1984			
1985	010670		ASCNST: .BLKW 5
			:RETURN
			:CONVERSION STORAGE AREA

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3


```

1987
1988
1989
1990
1991
1992 010702
1993 010702 012700 000120
1994 010706 012701 001603
1995 010712 122741 000040
1996 010716 001003
1997 010720 005300
1998 010722 001373
1999 010724 000436
2000
2001 010726
010726 005046
010730 005046
010732 005046
010734 005046
010736 010046
010740 012746 001463
010744 005046
010746 012746 000006
010752 005046
010754 112716 000001
010760 012746 000001
010764 012746 000000G
010770 012746
010772 003 014
010774 104377
010776 103002
011000 004757 000016
2002
2003
2004 011004 012701 001465
2005 011010 112721 000040
2006 011014 005300
2007 011016 001374
2008 011020 000207
2009
2010 011022
011022 012746 000000
011026 017646 000000
011032 062766 000002 000002
011040 017666 000002 000002
011046 012746
011050 123 003
011052 104377
;
;
; WRITE A PRINT LINE TO TT0
;
;
CONSOL:
MOV #00,R0 ;PRINT BUFFER BYTE COUNT
MOV #PRINT+78,R1 ;POINT PAST END OF BUFFER
1$: CMPB #40,-(R1) ;LOOK FOR A NON-BLANK
BNE 2$ ;OK, WRITE LINE
DEC R0 ;DEC CHAR COUNT
BNE 1$
BR ABEND2 ;NO NON-BLANKS?
;
2$: QIOW$S #IO,WVB,#LUN.TT,#EFN.1,STAT,<#PRINT-2,R0>,ABEND2
CLR -(SP)
CLR -(SP)
CLR -(SP)
CLR -(SP)
MOV R0,-(SP)
MOV #PRINT-2,-(SP)
CLR -(SP)
MOV #STAT,-(SP)
CLR -(SP)
MOV #EFN.1,(SP)
MOV #LUN.TT,-(SP)
MOV #IO,WVB,-(SP)
MOV (PC)+,-(SP)
.BYTE 3,12
EML +0<377>
BCC .+6
JSR PC,ABEND2
;
;
4$: MOV #PRINT,R1 ;POINT TO STRING
MOV #40,(R1)+ ;CLEAR LINE TO BLANKS
DEC R0 ;DEC LOOP COUNT
BNE 4$
RTS PC
;
ABEND2: ABRT$S #MYSELF
MOV #MYSELF,-(SP)
MOV 0(SP),-(SP)
ADD #2,2(SP)
MOV 02(SP),2(SP)
MOV (PC)+,-(SP)
.BYTE 03,3
EML +0<377>

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

2012.      ;
2013.      ;
2014.      ;      WRITE TO TT0 AND PROMPT.
2015.      ;
2016.      ;
2017 011054 005267 166732  ENDTST: INC.  ERWORD.
2018 011060 005267 166726  HLTST: INC.  ERWORD.
2019 011064 005267 166722  STOP: INC.  ERWORD.
2020 011070 005267 166716  BEGTST: INC.  ERWORD.
2021 011074 005267 166712  OUT1: INC.  ERWORD.
2022 011100 005267 166706  ERR12: INC.  ERWORD.
2023 011104 005267 166702  ERR11: INC.  ERWORD.
2024 011110 005267 166676  ERR10: INC.  ERWORD.
2025 011114 005267 166672  ERR9: INC.  ERWORD.
2026 011120 005267 166666  ERR8: INC.  ERWORD.
2027 011124 005267 166662  ERR7: INC.  ERWORD.
2028 011130 005267 166656  ERR6: INC.  ERWORD.
2029 011134 005267 166652  ERR5: INC.  ERWORD.
2030 011140 005267 166646  ERR4: INC.  ERWORD.
2031 011144 005267 166642  ERR3: INC.  ERWORD.
2032 011150 005267 166636  ERR2: INC.  ERWORD.
2033.      ;
2034 011154      NESTOP:
2035 011154 005267 166632  EROPT: INC.  ERWORD.
2036 011160 005267 166626  LPTST: INC.  ERWORD.
2037 011164 005267 166622  USRTST: INC.  ERWORD.
2038 011170 005267 166616  SELTST: INC.  ERWORD.
2039 011174 005267 166612  PMPTT: INC.  ERWORD.
2040 011200 005267 166606  PMPTO: INC.  ERWORD.
2041 011204 005267 166602  PMPTI: INC.  ERWORD.
2042 011210 005267 166576  PMPTC: INC.  ERWORD.
2043 011214 005267 166572  PMPTB: INC.  ERWORD.
2044 011220 005267 166566  PMPTA: INC.  ERWORD.
2045 011224 005267 166562  MEMSEL: INC.  ERWORD.
2046 011230 005267 166556  ALLSEL: INC.  ERWORD.
2047      000014  NEST. =.  <.-NESTOP>/4
2048.      ;
2049.      ;      USE THE INDEX ERWORD TO COUNT UP FROM THE BOTTOM OF THE
2050.      ;      MESSAGE TABLE. FIND THE END OF THE MESSAGE FIRST, THEN
2051.      ;      THE BEGINNING, THEN GET THE LENGTH.
2052.      ;
2053 011234 010246      MOV.  R2, -(SP)      ;SAVE R2
2054 011236 016702 166550  MOV.  ERWORD, R2.      ;LOAD LOOP COUNT
2055 011242 012701 003271*  MOV.  #ASCIZ, R1      ;POINT TO END OF MESSAGE TABLE
2056 011246 105741      1$:  TSTB.  -(R1)      ;LOOK FOR END OF MESSAGE
2057 011250 001376      BNE.  1$
2058 011252 005302      DEC.  R2.      ;LOOP COUNT.
2059 011254 001374      BNE.  1$      ;BACK UP ANOTHER MESSAGE.
2060 011256 010100      MOV.  R1, R0      ;SAVE POINTER TO END OF MESSAGE.
2061 011260 105741      2$:  TSTB.  -(R1)      ;BACK UP TO BEGINNING OF MESSAGE.
2062 011262 001376      BNE.  2$
2063 011264 005201      INC.  R1      ;BUMP TO FIRST CHAR OF MESSAGE.
2064 011266 160100      SUB.  R1, R0      ;R0 NOW = MESSAGE LENGTH.
2065 011270 012602      MOV.  (SP)+, R2.
2066.      ;
2067 011272.      QIOW$S. #10, VVB, #LUN, TT, #EFN, 1, #STAT, <R1, R0>, ABEND
2068 011272. 005046      CLR.  (SP)

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

011274	005046		CLR	-(SP)	
011276	005046		CLR	-(SP)	
011300	005046		CLR	-(SP)	
011302	010046		MOV	R0, -(SP)	
011304	010146		MOV	R1, -(SP)	
011306	005046		CLR	-(SP)	
011310	012746	000006*	MOV	#STAT, -(SP)	
011314	005046		CLR	-(SP)	
011316	112716	000001	MOVB	#EFN.1, (SP)	
011322	012746	000001	MOV	#LUN.TT, -(SP)	
011326	012746	000000G	MOV	#ID.UVB, -(SP)	
011332	012746		MOV	(PC)+, -(SP)	
011334	003	014	.BYTE	3, 12	
011336	104377		EMT	+0<377>	
011340	103002		BCC	.+6	
011342	004767	000106	JSR	PC, ABEND	
2068					
2069	011346		CLEF\$S	#EFN.1	
	011346	012746	MOV	#EFN.1, -(SP)	
	011352	012746	MOV	(PC)+, -(SP)	
	011354	037	.BYTE	31, 2	
	011356	104377	EMT	+0<377>	
2070	011360	105767	TSTB	STAT	:GOOD RETURN
2071	011364	003433	BLE	ABEND	:NO
2072					
2073					
2074			ISSUE	GCML	
2075	011366	022767	000014	166416	
2076	011374	002424			
2077			CMF	#NEST.ERWORD	:PROMPT WITH MESSAGE
			BLT	TTX	:NO, JUST EXIT
2078	011376	012700	000032*		
2079	011402	012701	000051		
2080	011406	005020			
2081	011410	005301			
2082	011412	001375			
2083					
2084	011414		GCML\$	#GCMBLK	
	011414	012700	MOV	#GCMBLK, R0	
	011420	005060	CLR	G.PSDS(R0)	
	011424	004767	JSR	PC, GCML1	
2085	011430	103411	BCC	ABEND	
2086	011432	016067	000146	166514	
2087	011440	012767	000032*	166510	
2088	011446	005067	166340		
2089	011452	000207			
2090					
2091	011454		ABEND:	ABRT\$S	#MYSELF
	011454	012746	MOV	#MYSELF, -(SP)	
	011460	017646	MOV	@(SP), -(SP)	
	011464	062766	ADD	#2, 2(SP)	
	011472	017666	MOV	@2(SP), 2(SP)	
	011500	012746	MOV	(PC)+, -(SP)	
	011502	123	.BYTE	83, 3	
	011504	104377	EMT	+0<377>	
2092					
2093	003630*		.END	START	

TMT-----MACRO:M1110 27-MAR-80 15:38 PAGE:28-2
SYMBOL TABLE

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

A.	= 000001	CK2.	001276RG	FN.FSC.	000004	004	GCMBLK.	003322R	MT10	000026R
ABEND	011454R	CK3	001300RG	FN.FSD.	000020	004	GCMBUF.	000032R	MYSELF.	000000R
ABEND2.	011022R	CMILUN.	000002	FN.MHR.	000010	004	GCMLN.	000154R	N.	= 000012
ALL	003634R	CNUM.	= 000006	FN.NMB.	000022	004	GCMFNT.	000156R	NEST.	= 000014
ALLSEL	011230R	CONSOL	010702R	FN.QLS.	000006	004	GE.BIF.	177775	NESTOP.	011154R
ALLSTR.	000401R	CTBL.	000430R	FN.RDC.	000014	004	GE.CLO.	000004	N.BFAC.	000004
ALLTST.	000010	CURLIM.	001150R	FN.UPD.	000012	004	GE.COM.	000001	N.BHGH.	000006
AMSG	001355R	DBSLEN.	000116	FSECK.	010404R		GE.CON.	000020	N.BTCH.	000004
ASCIZ	003271R	DG.ERR.	= 001000	FTBL	001417R		GE.EOF.	177766	N.BUFB.	004000
ASCNST	010670R	DG.SDF.	= 002000	F.ACTL.	= 000076		GE.IND.	000002	N.BUFW.	002000
AST	007206R	DG.TDF.	= 004000	F.ALOC.	= 000040		GE.IOR.	177777	N.DID.	= 000024
ASTWRD	000004R	DOUBLE	010540R	F.BBFS.	= 000062		GE.LC.	= 000010	N.DVNM.	000032
ASWRK	001316R	EAST	007564R	F.BDB.	= 000070		GE.MDE.	177774	N.FID.	= 000000
B.	= 000002	EFN.1	= 000001	F.BGBC.	= 000057		GE.OPR.	177776	N.FNAM.	000006
BASE.	000024R	EMSG.	001366R	F.BKDN.	= 000026		GE.RBG.	177730	N.FOS.	= 000764
BEGTST.	011070R	ENDLIM	004240R	F.BKDS.	= 000020		GE.SIZ.	000040	N.FTYP.	000014
BINWD.	000016R	ENDLN.	= 000014	F.BKEF.	= 000050		G.CMLD.	000146	N.FVER.	000016
BITVAL.	000000	ENDOF.	003306R	F.BKPI.	= 000051		G.DPRM.	000160	N.NEXT.	000022
BIT0	= 000001	ENDTST	011054R	F.BKST.	= 000024		G.ERN.	= 000140	N.QUERY.	000031
BIT1	= 000002	ERLIM.	000014R	F.BKVB.	= 000064		G.ISIZ.	000020	N.STAT.	000020
BIT10	= 002000	EROPT.	011154R	F.BKVS.	= 000075		G.LPDL.	000060	N.SUNT.	000002
BIT11	= 004000	ERPRMT	005160R	F.CNTG.	= 000034		G.MODE.	000141	N.UNIT.	000034
BIT12	= 010000	ERRADD	001304RG	F.DFNB.	= 000046		G.PSDS.	000142	O.	= 000020
BIT13	= 020000	ERRCT.	001306RG	F.DSPT.	= 000044		G.SIZE.	000224	OUT1	011074R
BIT14	= 040000	ERROR.	= 000040	F.DVNM.	= 000134		HALT.	= 000020	PACK	010412R
BIT15	= 100000	ERRORX.	007170R	F.EFBK.	= 000010		HLTTST.	011060R	PAR***	000027
BIT2	= 000004	ERR10	011110R	F.EFN.	= 000050		I.	= 000010	PASS	000162R
BIT3	= 000010	ERR11	011104R	F.EOB8.	= 000032		IO.ATA.	***** GX.	PASSC.	010510R
BIT4	= 000020	ERR12.	011100R	F.ERR.	= 000052		IO.UVB.	***** GX.	PASSH.	000160R
BIT5	= 000040	ERR2.	011150R	F.FACC.	= 000043		JMPMT.	005614R	PASSX.	010666R
BIT6	= 000100	ERR3	011144R	F.FFBY.	= 000014		LASTJ.	006036R	PAST	007600R
BIT7	= 000200	ERR4	011140R	F.FNAM.	= 000110		LASTJ1	006040R	PCLCX.	010444R
BIT8	= 000400	ERR5	011134R	F.FNB.	= 000102		LIMITS.	010002R	PLIM	004064R
BIT9	= 001000	ERR6	011130R	F.FTYP.	= 000116		LIMNUM.	000014	PMPTA.	011220R
BYTE0	= 000000	ERR7	011124R	F.FVER.	= 000120		LIMREF.	000700R	PMPTB.	011214R
BYTE1	= 000001	ERR8	011120R	F.HIBK.	= 000004		LIMX.	010164R	PMPTC.	011210R
BYTE2	= 000002	ERR9	011114R	F.LUN.	= 000042		LIMX2.	010200R	PMPTI.	011204R
BYTE3	= 000003	ERWORD.	000012R	F.MBCT.	= 000054		LOOP.	= 000001	PMPTO.	011200R
BYTE4	= 000004	ERW1	001310RG	F.MBC1.	= 000055		LOOPCT.	000164R	PMPTT.	011174R
BYTE5	= 000005	ERW2.	001312RG	F.MBFG.	= 000056		LOWER.	000022R	PMPT10	004532R
BYTE6	= 000006	ERW3	001314RG	F.NRBD.	= 000024		LPRMPT.	005032R	PMG	001330R
BYTE7	= 000007	FD.CCL.	***** GX.	F.NREC.	= 000030		LPTST.	011160R	PMG2.	003272R
BYTE8	= 000010	FD.FID	000000	F.OVBS.	= 000030	003	LUN.TT.	= 000001	PM2LN.	= 000014
BYTE9	= 000011	FD.FNB	000006	F.RACC.	= 000016	003	MEM.	= 000006	PREADD.	001302RG
BYTVAL.	000012	FD.FVR	000004	F.RATT.	= 000001	003	MEMERR.	006540RG	PRINT.	001465R
C.	= 000004	FD.LEN	000010	F.RCHM.	= 000034	003	MEMSEL.	011224R	PSECX.	010450R
CAST.	007460R	FD.REC.	***** GX.	F.RCTL.	= 000017		MEMTOP.	004012R	REST	007652R
CF.COT.	= 000041	FD.TTY.	***** GX.	F.RSIZ.	= 000002		MOVE.	003706R	RMSG	001400R
CF.DGN.	= 000046	FIND.	010274R	F.RTYP.	= 000000		MS.DGN.	= 010000	SCAN	001400R
CF.DHR.	= 000042	FMSG.	001335R	F.SEGN.	= 000100		MTCNT.	000172R	SELECT.	000166R
CF.DMC.	= 000047	FNIN1	010216R	F.SPDV.	= 000072		MTJUMP.	006530R	SELOOP.	004314R
CF.HBR.	= 000045	FNMTCH	010252R	F.SPUN.	= 000074		MTMAIN.	005400R	SELTST.	011170R
CF.HRL.	= 000044	FNOUT1	010210R	F.STBK.	= 000036		MTPNT.	000170R	SERR	004356R
CF.UPD.	= 000043	FNOUT2	010230R	F.UNIT.	= 000136		MTRF.	000460R	SETTBL.	000174R
CHECK.	004616R	FN.ACK	000016	F.URBD.	= 000020	004	MTRTN.	003704R	SR.ARE.	000114
CHECK0	005000R	FN.FSA	000000	F.VBN.	= 000064	004	MTSET.	005304R	SR.LARS	000106
CKDATA.	001274RG	FN.FSB	000002	F.VBSZ.	= 000060	004	MTSUB.	000730R	SR.DAY.	000010

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

TMT.....M1110 27-MAR-80 15:38 PAGE 28-3
SYMBOL TA

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

SR.DLT. 000014	002.S.FNAM= 000006	T\$BT. = 000020	T\$1CLK= 000400	UNMSG. 001412R
SR.ECB. 000047	002.S.FNB= 000036	T\$BTAR= 000030	T\$BBEN= 000020	UNPK. 010454R
SR.ECH. 000046	002.S.FNBW= 000017	T\$BTD= 002000	T1 006064R	UPPER. 000020R
SR.ECL. 000050	002.S.FNTY= 000004	T\$CD. = 000100	T1ADDR. 001214R	USRTST. 011164R
SR.FIB. 000012	002.S.FTYP= 000002	T\$CLK= 002000	T1F. = ***** GX	WAST. 007306R
SR.GRE. 000100	002.S.NFEN= 000020	T\$DISK= 000200	T1I. = ***** GX	WORD0= 000000
SR.GRS. 000072	002.T. = 000040	T\$DRD= 000004	T1O. = ***** GX	WORD1= 000002
SR.LEN. 000122	002.TAST. 007502R	T\$EMEM= 010000	T1T. = ***** GX	WORD2= 000004
SR.LIN. 000066	002.TCDADD 001244R	T\$FSAA= 000000	T10 006274R	WORD3= 000006
SR.LIP. 000062	002.TCUADD 001260R	T\$FSAB= 000004	T11 006300R	WORD4= 000010
SR.MDN. 000006	002.TD\$CTR= 176370	T\$FSAC= 000014	T12. 006424R	WORD5= 000012
SR.NDC. 000042	002.TD\$CTW= 176360	T\$FSB2= 000010	T12FD= ***** GX	WORD6= 000014
SR.NDS. 000036	002.TD\$INL= 004000	T\$IB. = 000026	T12FU= ***** GX	WORD7= 000016
SR.NIN. 000030	002.TD\$MEM= 000270	T\$IBAR= 000024	T12ID= ***** GX	WORD8= 000020
SR.NIP. 000022	002.TD\$QAR= 176344	T\$IBE. = 020000	T12IU= ***** GX	WORD9= 000022
SR.SDB. 000032	002.TD\$OTR= 176346	T\$IBF= 040000	T12OD= ***** GX	WRDVAL= 000024
SR.SRC. 000002	002.TD\$QRD= 000274	T\$ICD= 000040	T12OU= ***** GX	ZREL 004402R
SR.SUN. 000000	002.TD\$RST= 176366	T\$MODE= 004000	T12TD= ***** GX	\$CBDSG= ***** GX
SR.TWS. 000056	002.TD\$SW= 176376	T\$OB. = 000036	T12TU= ***** GX	\$CBOMG= ***** GX
SR.WSL. 000052	002.TD\$TAR= 176372	T\$OBE= 004000	T2. 006074R	\$CDTB= ***** GX
SR.YR. 000004	002.TD\$TAW= 176362	T\$OBF= 010000	T3 006110R	\$CQTB= ***** GX
SR.11N. 000024	002.TD\$TDR= 176374	T\$OBRA= 000034	T4 006126R	\$DDIV= ***** GX
SR.11P. 000016	002.TD\$TDW= 176364	T\$OBWA= 000032	T5 006144R	\$DIV= ***** GX
STADDR. 001200R	TESTS= 000014	T\$OUTA= 100000	T6 006174R	\$MUL= ***** GX
START. 003630R	TEST10= 000004	T\$RBDQ= 000200	T6CHK. 004574R	\$\$\$= 003502R
STAT. 000006R	TEST6= 000002	T\$RNB= 000040	T6FACT. 000030R	\$\$\$ARG= 000002
STOP. 011064R	TMSG. 001323R	T\$RSET= 040000	T7 006200R	\$\$\$T1= 000067
STRLEN= 000027	TROCT. 000202R	T\$SC. = 000022	T7ADDR. 001230R	\$\$\$T2= 000027
STUFF= ***** GX	TRT. 004344R	T\$SCLK= 020000	T7F. = ***** GX	FSRCB= ***** G
STUF1= ***** GX	TTX. 011446R	T\$SEG1= 000000	T7I. = ***** GX	GEMLI= ***** G
STUF0= ***** GX	T\$AD= 000020	T\$SEG2= 000001	T7O. = ***** GX	...PC1= 003322R
STUFT= ***** GX	T\$BA= 000002	T\$SEG3= 000002	T7T. = ***** GX	...PC2= 003524R
S.BFHD= 000020	T\$BD= 000010	T\$SO= 000001	T8 006210R	...PC3= 003322R
S.FATT= 000016	T\$BSO= 100000	T\$UBUS= 100000	T9 006240R	...TPC= 000020
S.FDB= 000140				

. ABS. 000000 000
011506 001
SRCOFF. 000122 002
FDSCOF. 000010 003
FNOFFS. 000022 004
\$\$FSR1 001020 005
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 7110 WORDS (28 PAGES)
DYNAMIC MEMORY: 8084 WORDS (31 PAGES)
ELAPSED TIME: 00:01:25
TMT, TMT, SP=C20, 13P, C, TMT

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
1
2 000000 .TITLE- TTEST-
3 .PSECT- TTEST-
4 .LIST- MEB-
5
6
7
8
9
10
11
12
13
14
15
16
17 000000 STUFF:
18 000000 016667 000002 000000G MOV 2(SP),PREADD-
19 000006 016767 000000G 176362 1$: MOV PREADD,TD$TAW-
20 000014 004767 000474 JSR PC,WRITE-
21 000020 005267 000000G INC PREADD-
22 000024 026667 000004 000000G CMP 4(SP),PREADD- ;CHECK FOR UPPER LIMITS-
23 000032 103365 BHIS 1$
24
25 000034 016667 000002 000000G MOV 2(SP),PREADD-
26 000042 016667 000002 176362 MOV 2(SP),TD$TAW-
27 000050 004767 000464 2$: JSR PC,READ-
28 000054 005267 000000G INC PREADD-
29 000060 026667 000004 000000G CMP 4(SP),PREADD- ;CHECK FOR UPPER LIMITS-
30 000066 103370 BHIS 2$
31 000070 000207 RTS PC-
32
33
34
35
36
37
38 000072 TIF:
39 000072 016667 000002 000000G MOV 2(SP),CKDATA- ;ADDRESS IS TEST PATTERN-
40 000100 016667 000002 000000G MOV 2(SP),PREADD-
41 000106 016767 000000G 176362 1$: MOV PREADD,TD$TAW-
42 000114 004767 000374 JSR PC,WRITE-
43 000120 005267 000000G INC CKDATA-
44 000124 005267 000000G INC PREADD-
45 000130 026667 000004 000000G CMP 4(SP),PREADD- ;CHECK FOR UPPER LIMITS-
46 000136 103363 BHIS 1$
47
48 000140 016667 000002 000000G MOV 2(SP),CKDATA- ;ADDRESS IS TEST PATTERN-
49 000146 016667 000002 000000G MOV 2(SP),PREADD-
50 000154 016667 000002 176362 MOV 2(SP),TD$TAW-
51 000162 004767 000352 2$: JSR PC,READ-
52 000166 005267 000000G INC CKDATA-
53 000172 005267 000000G INC PREADD-
54 000176 026667 000004 000000G CMP 4(SP),PREADD- ;CHECK FOR UPPER LIMITS-
55 000204 103366 BHIS 2$
56 000206 000207 RTS PC-
57
```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

58      ;
59      ;      TEST-06
60      ;      MEMORY CROSS-TALK TEST
61      ;
62      ;
63 000210      T6F:
64      ;
65      ;      MOV.      #10,R2.      ;SET LOOP COUNT.
66      ;      MOV.      #-1,CKDATA. ;SET TEST PATTERN TO ALL 1'S
67      ;      MOV.      2(SP),PREADD. ;SET UP START ADDRESS.
68      ;      MOV.      PREADD,TD$TAW. ;MOVE ADDR TO TRANSFER REG.
69      ;      JSR.      PC,WRITE.      ;WRITE 'CKDATA' TO 3 SEGS.
70      ;      ADD.      #2,PREADD.      ;SKIP AN ADDRESS.
71      ;      CMP.      4(SP),PREADD.   ;FINISHED.
72      ;      BHIS.     1$.             ;NO, CONTINUE.
73      ;      DEC.      R2.             ;SUB FROM LOOP COUNT.
74      ;      BNE.     10$.
75      ;
76      ;      CLR.      CKDATA.         ;SET TEST PATTERN TO ZERO.
77      ;      MOV.      2(SP),PREADD.   ;RESET START ADDRESS.
78      ;      INC.      PREADD.         ;START AT FIRST ADDR + 1
79      ;      MOV.      PREADD,TD$TAW.   ;MOVE START ADDRESS TO TRANS. REG.
80      ;      JSR.      PC,READ.        ;READ AND COMPARE.
81      ;      ADD.      #2,PREADD.      ;BUMP ADDR.
82      ;      CMP.      4(SP),PREADD.   ;FINISHED.
83      ;      BHIS.     2$.             ;NO.
84      ;      RTS.      PC.
85      ;
86      ;
87      ;      TEST-07
88      ;      WRITE/READ ALL ADDRESSES WITH ADDRESS COMPLEMENT.
89      ;
90 000212      T7F:
91 000212. 016602. 000002.      MOV.      2(SP),R2.      ;LOAD ADDRESS.
92 000216. 016667. 000002. 000000G. MOV.      2(SP),PREADD. ;SET UP START ADDRESS.
93 000224. 005102.      COM.      R2.      ;GET ADDRESS COMPLEMENT.
94 000226. 010267. 000000G.      MOV.      R2,CKDATA.      ;SET TEST PATTERN.
95 000232. 016767. 000000G. 176362. MOV.      PREADD,TD$TAW. ;MOVE ADDR TO TRANSFER REG.
96 000240. 004767. 000250.      JSR.      PC,WRITE.      ;WRITE 'CKDATA' TO 3 SEGS.
97 000244. 005267. 000000G.      INC.      PREADD.      ;BUMP TO NEXT ADDRESS.
98 000250. 016702. 000000G.      MOV.      PREADD,R2.      ;SET R2 TO NEXT ADDRESS.
99 000254. 026667. 000004. 000000G. CMP.      4(SP),PREADD. ;FINISHED.
100 000262. 103360.      BHIS.     1$.             ;NO, CONTINUE.
101      ;
102 000264. 016667. 000002. 176362. MOV.      2(SP),TD$TAW. ;MOVE START ADDRESS TO TRANS REG.
103 000272. 016602. 000002.      MOV.      2(SP),R2.      ;SET R2 = ADDRESS.
104 000276. 016667. 000002. 000000G. MOV.      2(SP),PREADD. ;RESET START ADDRESS.
105 000304. 005102.      COM.      R2.      ;GET ADDRESS COMPLEMENT.
106 000306. 010267. 000000G.      MOV.      R2,CKDATA.      ;SET TEST PATTERN.
107 000312. 004767. 000222.      JSR.      PC,READ.      ;READ AND COMPARE.
108 000316. 005267. 000000G.      INC.      PREADD.      ;BUMP ADDR.
109 000322. 016702. 000000G.      MOV.      PREADD,R2.      ;SET R2 = NEXT ADDRESS.
110 000326. 026667. 000004. 000000G. CMP.      4(SP),PREADD. ;FINISHED.
111 000334. 103363.      BHIS.     2$.             ;NO.
112 000336. 000207.      RTS.      PC.
113      ;
114      ;

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
115      :      TEST-12.
116      :      ADDRESSING TEST.
117      :
118      :
119      000340      T12FD::
120      000340      016667      000002      000000G      MOV.      2(SP),PREADD.      :GET START ADDRESS
121      000346      016767      000000G      000000G      1$:      MOV.      CK2,CKDATA      :GET READ PATTERN
122      000354      016767      000000G      176362      MOV.      PREADD,TD$TAW
123      000362      004767      000152      JSR.      PC,READ
124      000366      016767      000000G      000000G      MOV.      CK3,CKDATA      :GET WRITE PATTERN
125      000374      016767      000000G      176362      MOV.      PREADD,TD$TAW
126      000402      004767      000106      JSR.      PC,WRITE
127      000406      062767      000001      000000G      ADD.      #1,PREADD
128      000414      026667      000004      000000G      CMP.      4(SP),PREADD
129      000422      103351      BHIS.      1$
130      000424      000207      RTS.      PC
131      :
132      000426      T12FU::
133      000426      016667      000004      000000G      MOV.      4(SP),PREADD      :START ADDRESS = UPPER LIMIT
134      000434      016767      000000G      000000G      1$:      MOV.      CK2,CKDATA      :READ PATTERN
135      000442      016767      000000G      176362      MOV.      PREADD,TD$TAW
136      000450      004767      000064      JSR.      PC,READ
137      000454      016767      000000G      000000G      MOV.      CK3,CKDATA      :WRITE PATTERN
138      000462      016767      000000G      176362      MOV.      PREADD,TD$TAW
139      000470      004767      000020      JSR.      PC,WRITE
140      000474      162767      000001      000000G      SUB.      #1,PREADD
141      000502      026667      000002      000000G      CMP.      2(SP),PREADD
142      000510      003751      BLE.      1$
143      000512      000207      RTS.      PC
144      :
145      :
146      :
147      :
148      :      WRITE SUBROUTINE...
149      :
150      000514      016767      000000G      176364      WRITE:      MOV.      CKDATA,TD$TDW
151      000522      016767      000000G      176364      MOV.      CKDATA,TD$TDW
152      000530      016767      000000G      176364      MOV.      CKDATA,TD$TDW
153      000536      000207      RTS.      PC
154      :
155      :
156      :
157      :      READ SUBROUTINE...
158      000540      READ:
159      000540      010146      SAVE.      R1,R2,R3
160      000542      010246      MOV.      R1,-(SP)
161      000544      010346      MOV.      R2,-(SP)
162      000546      016701      176374      MOV.      TD$TDR,R1
163      000552      020167      000000G      CMP.      R1,CKDATA
164      000556      004767      000100      BEQ.      1$
165      000564      000424      JSR.      PC,PRPERR
166      000566      016701      176374      BR.      READX
167      000572      020167      000000G      1$:      MOV.      TD$TDR,R1
168      000576      001403      CMP.      R1,CKDATA
169      000578      001403      BEQ.      2$
```



```

169 000600 004767 000060 JSR PC,PRPERR.
170 000604 000414 BR READX.
171 000606 016702 176374 2$: MOV TD$TDR,R2.
172 000612 016703 000000G MOV CKDATA,R3.
173 000616 042702 000777 BIC #000777,R2. ;CLEAR 9 BITS.
174 000622 042703 000777 BIC #000777,R3.
175 000626 020203 CMP R2,R3.
176 000630 001402 BEQ READX.
177 000632 004767 JSR PC,PRPERR.
178
179 000636 ;
000636 012603 READX: RESTOR R1,R2,R3.
000640 012602 MOV (SP)+,R3.
000642 012601 MOV (SP)+,R2.
180 000644 000207 MOV (SP)+,R1.
181 RTS PC.
182
183 ;
184 ; READ WITHOUT COMPARE.
185 ;
186 000646 READ1:
187 000646 016746 176374 MOV TD$TDR,-(SP)
188 000652 016716 176374 MOV TD$TDR,(SP)
189 000656 016726 176374 MOV TD$TDR,(SP)+
190 000662 000207 RTS PC.
191
192 ;
193 ; PRE@ARE TO CALL ERROR ROUTINE.
194 ;
195 000664 PRPERR:
196 000664 016767 000000G-000000G MOV PREADD,ERRADD.
197 000672 016767 000000G 176362 MOV ERRADD,TD$TAW. ;SET ERR ADDRESS.
198 000700 012767 000003 000000G MOV #3,ERRCT. ;SEGMENT COUNT.
199 000706 016767 176374 000000G MOV TD$TDR,ERW1. ;READ SEGMENT 0.
200 000714 016767 176374 000000G MOV TD$TDR,ERW2. ;READ SEGMENT 1.
201 000722 016767 176374 000000G MOV TD$TDR,ERW3. ;READ SEGMENT 2.
202 000730 042767 000777 000000G BIC #000777,ERW3.
203
204 ;
205 ; RETRY READ.
206 000736 012746 000005 MOV #5,-(SP) ;NUMBER OF RETRIES
207 000742 016767 000000G 176362 RR1: MOV ERRADD,TD$TAW.
208 000750 016746 176374 MOV TD$TDR,-(SP) ;READ SEG 1
209 000754 016746 176374 MOV TD$TDR,-(SP) ;SEG 2.
210 000760 016746 176374 MOV TD$TDR,-(SP) ;SEG 3
211 000764 042716 000777 BIC #000777,(SP)
212 000770 022667 000000G CMP (SP)+,ERW3 ;SAME AS PREVIOUS.
213 000774 001012 BNE 1$ ;NO, ERROR UNPREDICTABLE.
214 000776 022667 000000G CMP (SP)+,ERW2 ;SAME AS PREVIOUS.
215 001002 001011 BNE 2$ ;NO.
216 001004 022667 000000G CMP (SP)+,ERW1 ;SAME.
217 001010 001010 BNE 3$ ;NO.
218 001012 005316 DEC (SP) ;SUB FROM # RETRIES.
219 001014 001352 BNE RR1
220 001016 005726 TST (SP)+
221 001020 000411 BR CALLER. ;CALL ERROR REPORTING ROUTINE.
222

```

TTEST...M1110 27-MAR-80 15:41 PAGE 8-4

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

223	001022	062706	000002	1\$:	ADD	#2,SP	
224	001026	062706	000002	2\$:	ADD	#2,SP	
225	001032	062706	000002	3\$:	ADD	#2,SP	
226	001036	012767	177777	000000G	MOV	#-1,ERRCT	; INDICATE UNPREDICTABLE RESULTS
227							
228	001044	004767	000000G	CALLER:	JSR	PC, MEMERR	
229	001050	000207			RTS	PC	

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
231      :  
232      :  
233      :      INPUT BUFFER TESTS.  
234      :  
235      :  
236      :      WRITE AND READ THE TEST PATTERN.  
237      :  
238 001052.      STUFF1:  
239 001052. 016667 000002 000000G.      MOV. 2(SP),PREADD.  
240 001060 016701 000000G.      1$: MOV. PREADD,R1  
241 001064 016711 000000G.      MOV. CKDATA,(R1)  
242 001070 012767 000000 176376      MOV. #0,TD$SW.  
243 001076 062767 000002 000000G.      ADD. #2,PREADD.  
244 001104 026667 000004 000000G.      CMP. 4(SP),PREADD.  
245 001112. 103362.      BHIS. 1$  
246      :  
247 001114 016667 000002 000000G.      MOV. 2(SP),PREADD.  
248 001122. 004767 000416 000000G.      2$: JSR. PC,READI.      ;READ AND COMPARE  
249 001126 026667 000004. 000000G.      CMP. 4(SP),PREADD.  
250 001134 103372.      BHIS. 2$  
251 001136 000207      RTS. PC.  
252      :  
253      :  
254      :      TEST-01  
255      :      WRITE ADDRESS INTO ITSELF.  
256      :  
257      :  
258 001140      T11:  
259 001140 016667 000002. 000000G.      MOV. 2(SP),PREADD.  
260 001146 016767 000000G. 000000G. 1$: MOV. PREADD,CKDATA.  
261 001154 016701 000000G.      MOV. PREADD,R1  
262 001160 016711 000000G.      MOV. CKDATA,(R1)  
263 001164 012767 000000 176376      MOV. #0,TD$SW.  
264 001172. 062767 000002 000000G.      ADD. #2,PREADD.  
265 001200 026667 000004. 000000G.      CMP. 4(SP),PREADD.      ;CHECK FOR UPPER LIMITS  
266 001206 103357      BHIS. 1$  
267      :  
268 001210 016667 000002 000000G.      MOV. 2(SP),PREADD.  
269 001216 016767 000000G. 000000G. 2$: MOV. PREADD,CKDATA.      ;ADDRESS IS PATTERN  
270 001224 004767 000314 000000G.      JSR. PC,READI.  
271 001230 026667 000004. 000000G.      CMP. 4(SP),PREADD.      ;CHECK FOR UPPER LIMITS  
272 001236 103367      BHIS. 2$  
273 001240 000207      RTS. PC.  
274      :  
275      :  
276      :      TEST-06  
277      :      MEMORY CROSS-TALK TEST  
278      :  
279      :  
280 001242.      T6I:  
281      :      MOV. #10.,R2.      ;SET LOOP COUNT.  
282      :      MOV. #-1,CKDATA.      ;SET TEST PATTERN TO ALL 1'S  
283      : 10$: MOV. 2(SP),PREADD.      ;SET UP START ADDRESS.  
284      : 1$: MOV. PREADD,R1      ;LOAD ADDRESS.  
285      :      MOV. CKDATA,(R1)  
286      :      MOV. #0,TD$SW.  
287      :      ADD. #4,PREADD.      ;SKIP AN ADDRESS.
```

```

288      :      CMP.      4(SP),PREADD.      :FINISHED.
289      :      BHIS.      1$      :NO, CONTINUE.
290      :      DEC.      R2.      :SUB FROM LOOP COUNT.
291      :      BNE.      10$
292      :
293      :
294      :      CLR.      CKDATA.      :SET TEST PATTERN TO ZERO.
295      :      MOV.      2(SP),PREADD.      :RESET START ADDRESS.
296      :      ADD.      #2,PREADD.
297      :2$:      JSR.      PC,READ1.      :READ AND COMPARE.
298      :      ADD.      #2,PREADD.      :BUMP TO NEXT ADDR (SKIP ONE)
299      :      CMP.      4(SP),PREADD.      :FINISHED.
300      :      BHIS.      2$      :NO.
301      :      RTS.      PC.
302      :
303      :
304      :      TEST 07
305      :      WRITE/READ ALL ADDRESSES WITH ADDRESS COMPLEMENT
306      :
307      :      T71::
308      :      MOV.      2(SP),R2.      :SET R2 = ADDRESS
309      :      MOV.      2(SP),PREADD.      :SET UP START ADDRESS.
310      :      COM.      R2.      :GET ADDRESS COMPLEMENT.
311      :      MOV.      R2,CKDATA.      :SET TEST PATTERN = ADDRESS COMPLEMENT
312      :      PREADD,R1      :LOAD ADDRESS.
313      :      MOV.      CKDATA,(R1)
314      :      MOV.      #0,TD$SW.
315      :      ADD.      #2,PREADD.
316      :      MOV.      PREADD,R2.      :SET R2 = ADDRESS
317      :      CMP.      4(SP),PREADD.      :FINISHED.
318      :      BHIS.      1$      :NO, CONTINUE.
319      :
320      :
321      :      MOV.      2(SP),R2.      :SET R2 = ADDRESS
322      :      MOV.      2(SP),PREADD.      :RESET START ADDRESS.
323      :      COM.      R2.      :GET ADDRESS COMPLEMENT.
324      :      MOV.      R2,CKDATA.      :SET TEST PATTERN = ADDR COM
325      :      JSR.      PC,READ1.      :READ AND COMPARE.
326      :      PREADD,R2.      :SET R2 = NEXT ADDRESS
327      :      CMP.      4(SP),PREADD.      :FINISHED.
328      :      BHIS.      2$      :NO.
329      :      RTS.      PC.
330      :
331      :
332      :      TEST 12:
333      :      ADDRESSING TEST.
334      :
335      :      T121D::
336      :      MOV.      2(SP),PREADD.      :GET START ADDRESS
337      :      MOV.      PREADD,R1      :LOAD READ ADDRESS
338      :      MOV.      (R1),R1      :READ DATA.
339      :      CMP.      CK2,R1      :COMPARE AGAINST PATTERN.
340      :      BEQ.      .+6
341      :      JSR.      PC,PRPER1
342      :
343      :
344      :      MOV.      PREADD,R1
345      :      MOV.      CK3,(R1)      :WRITE PATTERN

```

```

345 001424 012767 000000 176376      MOV.    #0,TD$SW.
346 001432 062767 000002 000000G.    ADD.    #2,PREADD.
347 001440 026667 000004 000000G.    CMP.    4(SP),PREADD.
348 001446 103352.                    BHIS.   1$
349 001450 000207.                    RTS.    PC.
350
351 001452.                    ;
352 001452. 016667 000004 000000G.    Ti2IU:: MOV.    4(SP),PREADD. ;START ADDRESS = UPPER LIMIT
353 001460 016701 000000G.    1$:      MOV.    PREADD,R1
354 001464 011101.                    MOV.    (R1),R1
355 001466 012767 000000 176376      MOV.    #0,TD$SW.
356 001474 026701 000000G.    CMP.    CK2,R1
357 001500 001402.                    BEQ.    .+6
358 001502. 004767 000074.                    JSR.    PC,PRPER1
359
360 001506 016701 000000G.                    MOV.    PREADD,R1
361 001512 016711 000000G.                    MOV.    CK3,(R1) ;WRITE PATTERN
362 001516 012767 000000 176376      MOV.    #0,TD$SW.
363 001524 152767 000002 000000G.    SUB.    #2,PREADD. ;BACK UP
364 001532 026667 000002 000000G.    CMP.    2(SP),PREADD.
365 001540 101747.                    BLOS.   1$
366 001542. 000207.                    RTS.    PC.
367
368
369
370
371
372 001544 016701 000000G.                    READ: MOV.    PREADD,R1
373 001550 011101.                    MOV.    (R1),R1
374 001552 012767 000000 176376      MOV.    #0,TD$SW.
375 001560 026701 000000G.    CMP.    CKDATA,R1
376 001564 001402.                    BEQ.    .+6
377 001566 004767 000010.                    JSR.    PC,PRPER1
378 001572 062767 000002 000000G.    ADD.    #2,PREADD.
379 001600 000207.                    RTS.    PC.
380
381
382
383
384
385 001602.                    PRPER1:
386 001602.                    SAVE.   R1,R2,R3
387 001602. 010146.                    MOV.    R1,-(SP)
388 001604 010246.                    MOV.    R2,-(SP)
389 001606 010346.                    MOV.    R3,-(SP)
390
391 001610 016767 000000G. 000000G.    MOV.    PREADD,ERRADD.
392 001616 012767 000001. 000000G.    MOV.    #1,ERRCT
393 001624 016701 000000G.    MOV.    PREADD,R1
394 001630 011167 000000G.    MOV.    (R1),ERW1
395
396
397
398
399 001634 012702. 000005.                    RETRY: MOV.    #5,R2
400 001640 011103.                    1$:      MOV.    (R1),R3 ;NUMBER OF RETRIES
401 001642. 026703 000000G.    CMP.    ERW1,R3 ;GET A WORD FROM MEMORY
402 001646 001003.                    BNE.    2$

```

TTEST...M 00-M1110 27-MAR-80 15:41 PAGE 9-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

399	001650	005302		DEC	... R2	
400	001652	001372		BNE	1\$	
401	001654	000403		BR	3\$	
402			:			
403	001656	012767	177777 000000G	2\$:	MOV	#-1,ERRCT
404	001664	004767	000000G	3\$:	JSR	PC, MEMERR
405			:			: INDICATE UNPREDICTABLE RESULTS
406	001670					
	001670	012603		RESTOR	R1, R2, R3	
	001672	012602		MOV	(SP)+, R3	
	001674	012601		MOV	(SP)+, R2	
407	001676	000207		MOV	(SP)+, R1	
				RTS	PC	

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
409      ;
410      ;
411      ;
412      ;
413      ;
414      ;
415      ;
416      ;
417 001700      STUFO:
418 001700      012767 000000 176376      MOV.  #0,TD$SW.      ;RESET TERM DETECTOR
419 001706      016667 000002 000000G.      MOV.  2(SP),PREADD.
420 001714      004767 000420      1$:      JSR.  PC,WRITE0.      ;WRITE OUTPUT BUFFER
421 001720      026667 000004 000000G.      CMP.  4(SP),PREADD.      ;CHECK FOR UPPER LIMITS
422 001726      103372      BHIS.  1$
423      ;
424 001730      016667 000002 000000G.      MOV.  2(SP),PREADD.
425 001736      004767 000436      2$:      JSR.  PC,READ0.      ;READ AND COMPARE
426 001742      026667 000004 000000G.      CMP.  4(SP),PREADD.      ;CHECK FOR UPPER LIMITS
427 001750      103372      BHIS.  2$
428 001752      000207      RTS.  PC
429      ;
430      ;
431      ;
432      ;
433      ;
434      ;
435 001754      T10:
436 001754      012767 000000 176376      MOV.  #0,TD$SW.      ;RESET TERM DETECTOR
437 001762      016667 000002 000000G.      MOV.  2(SP),CKDATA.      ;ADDRESS IS TEST PATTERN
438 001770      016667 000002 000000G.      MOV.  2(SP),PREADD.
439 001776      004767 000336      1$:      JSR.  PC,WRITE0.      ;WRITE OUTPUT BUFFER
440 002002      062767 000002 000000G.      ADD.  #2,CKDATA.      ;BUMP TO NEXT PATTERN (ADDRESS)
441 002010      026667 000004 000000G.      CMP.  4(SP),PREADD.      ;CHECK FOR UPPER LIMITS
442 002016      103367      BHIS.  1$
443      ;
444 002020      016667 000002 000000G.      MOV.  2(SP),CKDATA.      ;SET TEST PATTERN = ADDRESS
445 002026      016667 000002 000000G.      MOV.  2(SP),PREADD.
446 002034      004767 000340      2$:      JSR.  PC,READ0.      ;READ AND COMPARE
447 002040      062767 000002 000000G.      ADD.  #2,CKDATA.      ;CHANGE ADDRESS (PATTERN)
448 002046      026667 000004 000000G.      CMP.  4(SP),PREADD.      ;CHECK FOR UPPER LIMITS
449 002054      103367      BHIS.  2$
450 002056      000207      RTS.  PC
451      ;
452      ;
453      ;
454      ;
455      ;
456      ;
457 002060      T60:
458      ;
459      ;
460      ;
461      ;
462      ;
463      ;
464      ;
465      ;
```

```
466      : DEC ---- R2.          : SUB FROM LOOP COUNT.
467      : BNE 10$
468      :
469      : CLR CKDATA.          : SET TEST PATTERN TO ZERO.
470      : MOV 2(SP),PREADD.    : RESET START ADDRESS.
471      : ADD #2,PREADD.       : START AT NEXT ADDRESS.
472      :2$: JSR PC,READ0.    : READ AND COMPARE.
473      : ADD #2,PREADD.       : BUMP ADDR (SKIP ONE)
474      : CMP 4(SP),PREADD.   : FINISHED.
475      : BHIS 2$             : NO.
476 002060 000207      : RTS PC.
477      :
478      :
479      : TEST 07
480      : WRITE/READ ALL ADDRESSES WITH ADDRESS COMPLEMENT.
481      :
482      :
483 002062.      : T70::
484 002062. 012767 000000 176376      : MOV #0,TD$SW          : RESET TERM DETECTOR.
485 002070 016602 000002      : MOV 2(SP),R2.         : SET R2 = ADDRESS
486 002074 016667 000002 000000G.    : MOV 2(SP),PREADD      : SET UP START ADDRESS.
487 002102 005102.      : COM R2.              : GET ADDRESS COMPLEMENT.
488 002104 010267 000000G.    : MOV R2,CKDATA        : SET TEST PATTERN = ADDRESS COMPLEMENT
489 002110 004767 000224      : JSR PC,WRITE0.        : WRITE OUTPUT BUFFER.
490 002114 016702 000000G.    : MOV PREADD,R2.       : SET R2 = NEXT ADDRESS
491 002120 026667 000004 000000G.    : CMP 4(SP),PREADD.    : FINISHED.
492 002126 103365      : BHIS 1$              : NO, CONTINUE.
493      :
494 002130 016602 000002      : MOV 2(SP),R2.         : SET R2 = ADDRESS
495 002134 016667 000002 000000G.    : MOV 2(SP),PREADD      : RESET START ADDRESS.
496 002142 005102.      : COM R2.              : GET ADDRESS COMPLEMENT
497 002144 010267 000000G.    : MOV R2,CKDATA        : SET TEST PATTERN = ADDR.COM
498 002150 004767 000224      : JSR PC,READ0.        : READ AND COMPARE.
499 002154 016702 000000G.    : MOV PREADD,R2.       : SET R2 = NEXT ADDRESS
500 002160 026667 000004 000000G.    : CMP 4(SP),PREADD.    : FINISHED.
501 002166 103365      : BHIS 2$              : NO.
502 002170 000207      : RTS PC.
503      :
504      :
505      : TEST 12.
506      : ADDRESSING TEST.
507      :
508      :
509 002172.      : T120D::
510 002172. 012767 000000 176376      : MOV #0,TD$SW          : RESET TERM DETECTOR.
511 002200 016667 000002 000000G.    : MOV 2(SP),PREADD      : GET START ADDRESS
512 002206 016767 000000G 000000G. 1$: MOV CK2,CKDATA.        : GET READ PATTERN.
513 002214 004767 000100      : JSR PC,READ0.        : READ AND COMPARE.
514 002220 162767 000002 000000G.    : SUB #2,PREADD.       : WRITE AT SAME ADDRESS
515 002226 016767 000000G 000000G.    : MOV CK3,CKDATA.     : GET WRITE PATTERN.
516 002234 004767 000100      : JSR PC,WRITE0.       : WRITE OUTPUT BUFFER.
517 002240 026667 000004 000000G.    : CMP 4(SP),PREADD.    : FINISHED.
518 002246 103357      : BHIS 1$              : NO.
519 002250 000207      : RTS PC.
520      :
521 002252.      : T120U::
522 002252. 012767 000000 176376      : MOV #0,TD$SW          : RESET TERM DETECTOR.
```


Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

523 002260 016667 000004 000000G. MOV. 4(SP),PREADD. ;START ADDRESS = UPPER LIMIT
524 002266 016767 000000G.000000G.1$: MOV. CK2,CKDATA. ;READ PATTERN
525 002274 004767 000100 JSR. PC,READO. ;READ AND COMPARE
526 002300 162767 000002 000000G. SUB. #2,PREADD. ;BACK-UP TO SAME ADDRESS
527 002306 016767 000000G.000000G. MOV. CK3,CKDATA. ;WRITE PATTERN
528 002314 004767 000020 JSR. PC,WRITEO. ;WRITE OUTPUT BUFFER
529 002320 162767 000004 000000G. SUB. #4,PREADD. ;BACK-UP 1 ADDRESS
530 002326 026667 000002 000000G. CMP. 2(SP),PREADD.
531 002334 003754 BLE. 1$
532 002336 000207 RTS. PC
533
534
535
536
537
538
539 002340 012767 000032 176362. WRITEO: MOV. #T$OBWA,TD$TAW ;SET-UP MEMORY ADDRESS REG
540 002346 016767 000000G.176364 MOV. PREADD,TD$TDW.
541 002354 012767 000036 176362. MOV. #T$OB,TD$TAW. ;SET-UP TRANSFER CODE
542 002362 016767 000000G.176364 MOV. CKDATA,TD$TDW. ;WRITE OUTPUT BUFFER
543 002370 062767 000002 000000G. ADD. #2,PREADD.
544 002376 000207 RTS. PC
545
546
547
548
549
550 002400
551 002400 012767 000034 176362. READO: MOV. #T$OBRA,TD$TAW ;SET-UP READ ADDRESS
552 002406 016767 000000G.176364 MOV. PREADD,TD$TDW. ;ADDRESS TO READ
553 002414 012767 000036 176372. MOV. #T$OB,TD$TAR. ;SELECT OUTPUT BUFFER
554 002422 016701 176374 MOV. TD$TDR,R1 ;READ OUTPUT BUFFER
555 00242C 026701 000000G. CMP. CKDATA,R1
556 002432 001402 BEQ. +6 ;OK, CONTINUE
557 002434 004767 000010 JSR. PC,PRPERO.
558 002440 062767 000002 000000G. ADD. #2,PREADD. ;ADVANCE ADDRESS
559 002446 000207 RTS. PC
560
561
562
563
564
565 002450
566 002450 002450 PRPERO: SAVE. R1,R2,R3
567 002450 010146 MOV. R1,-(SP)
568 002452 010246 MOV. R2,-(SP)
569 002454 010346 MOV. R3,-(SP)
570
571 002456 016767 000000G.000000G. MOV. PREADD,ERRADD.
572 002464 012767 000001 000000G. MOV. #1,ERRCT. ;PRINT ONE WORD
573 002472 012767 000034 176362. MOV. #T$OBRA,TD$TAW ;SET-UP READ ADDRESS
574 002500 016767 000000G.176364 MOV. PREADD,TD$TDW. ;ADDRESS TO READ
575 002506 012767 000036 176372. MOV. #T$OB,TD$TAR. ;SELECT OUTPUT BUFFER
576 002514 016767 176374 000000G. MOV. TD$TDR,ERR1 ;READ OUTPUT BUFFER
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
577 002522 012702 000005      MOV.    #5,R2      ;NUMBER OF RETRIES
578 002526 012767 000034 176362 1$:  MOV.    #T$OBRA,TD$TAW ;SET UP READ ADDRESS
579 002534 016767 000000G 176364      MOV.    PREADD,TD$TDW  ;ADDRESS TO READ
580 002542 012767 000036 176372      MOV.    #T$OB,TD$TAR  ;SELECT OUTPUT BUFFER
581 002550 016703 176374      MOV.    TD$TDR,R3     ;READ OUTPUT BUFFER
582 002554 020301      CMP.    R3,R1
583 002556 001003      BNE.    2$          ;NOT THE SAME ERROR
584 002560 005302      DEC.    R2
585 002562 001361      BNE.    1$
586 002564 000403      BR.     3$          ;REPORT ERROR
587                                     ;
588 002566 012767 177777 000000G 2$:  MOV.    #1,ERRCT     ;INDICATE UNPREDICTABLE RESULTS
589 002574 004767 000000G 3$:  JSR.    PC,MEMERR
590                                     ;
591 002600      RESTOR. R1,R2,R3
      002600      MOV.    (SP)+,R3
      002602      MOV.    (SP)+,R2
      002604      MOV.    (SP)+,R1
592 002606 000207      RTS.    PC
```

```

594      ;
595      ;
596      ;      TEST ROUTINES FOR BYTE TRANSLATOR.
597      ;
598      ;
599      ;      THIS ROUTINE WRITES AND READS THE TEST PATTERN CONTAINED
600      ;      IN 'CKDATA'
601      ;
602      002610      STUT::
603      002610      012767      000000      176376      MOV.      #0,TD$SW      ;RESET TERM DETECTOR.
604      002616      016667      000002      000000G.      MOV.      2(SP),PREADD
605      002624      004767      000414      1$:      JSR.      PC,WRITET      ;WRITE BYTE TRANS
606      002630      026667      000004      000000G.      CMP.      4(SP),PREADD      ;CHECK FOR UPPER LIMITS
607      002636      103372      BHIS.      1$
608      ;
609      002640      016667      000002      000000G.      MOV.      2(SP),PREADD
610      002646      004767      000432      2$:      JSR.      PC,READT      ;READ AND COMPARE
611      002652      026667      000004      000000G.      CMP.      4(SP),PREADD      ;CHECK FOR UPPER LIMITS
612      002660      103372      BHIS.      2$
613      002662      000207      RTS.      PC
614      ;
615      ;
616      ;      TEST-01
617      ;      WRITE ADDRESS INTO ITSELF.
618      ;
619      ;
620      002664      T1T::
621      002664      012767      000000      176376      MOV.      #0,TD$SW      ;RESET TERM DETECTOR.
622      002672      016667      000002      000000G.      MOV.      2(SP),CKDATA      ;ADDRESS IS TEST PATTERN
623      002700      016667      000002      000000G.      MOV.      2(SP),PREADD
624      002706      004767      000332      1$:      JSR.      PC,WRITET      ;WRITE BYTE TRANS
625      002712      005267      000000G.      INC.      CKDATA
626      002716      026667      000004      000000G.      CMP.      4(SP),PREADD      ;CHECK FOR UPPER LIMITS
627      002724      103370      BHIS.      1$
628      ;
629      002726      016667      000002      000000G.      MOV.      2(SP),CKDATA      ;SET TEST PATTERN = ADDRESS
630      002734      016667      000002      000000G.      MOV.      2(SP),PREADD
631      002742      004767      000336      2$:      JSR.      PC,READT      ;READ AND COMPARE
632      002746      005267      000000G.      INC.      CKDATA
633      002752      026667      000004      000000G.      CMP.      4(SP),PREADD      ;CHECK FOR UPPER LIMITS
634      002760      103370      BHIS.      2$
635      002762      000207      RTS.      PC
636      ;
637      ;
638      ;      TEST-06
639      ;      MEMORY CROSS-TALK TEST
640      ;
641      ;
642      002764      T6T::
643      ;      MOV.      #0,TD$SW      ;RESET TERM DETECTOR.
644      ;      MOV.      #10,R2      ;SET LOOP COUNT.
645      ;      MOV.      #-1,CKDATA      ;SET TEST PATTERN TO ALL 1'S
646      ;      MOV.      2(SP),PREADD      ;SET UP START ADDRESS.
647      ;      10$:      JSR.      PC,WRITET      ;WRITE CKDATA TO MEMORY TRANS
648      ;      1$:      INC.      PREADD      ;SKIP AND ADDRESS.
649      ;      CMP.      4(SP),PREADD      ;FINISHED.
650      ;      BHIS.      1$      ;NO CONTINUE

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
651      :      DEC----- R2-      :SUB-FROM-LOOP-COUNT-
652      :      BNE-      10$
653      :
654      :      CLR-      CKDATA-      :SET-TEST-PATTERN-TO-ZERO-
655      :      MOV-      2(SP),PREADD- :RESET-START-ADDRESS-
656      :      INC-      PREADD-      :START-AT-FIRST-ADDR+ 1
657      :2$:      JSR-      PC,READT-      :READ-AND-COMPARE-
658      :      INC-      PREADD-      :BUMP-ADDR-(SKIP)
659      :      CMP-      4(SP),PREADD- :FINISHED-
660      :      BHIS-      2$      :NO-
661 002764 000207      RTS-      PC-
662      :
663      :
664      :      TEST-07
665      :      WRITE/READ-ALL-ADDRESSES-WITH-ADDRESS-COMPLEMENT-
666      :
667      :
668 002766      T7T::
669 002766 012767 000000 176376      MOV-      #0,TD$SW-      :RESET-TERM-DETECTOR-
670 002774 016602 000002      MOV-      2(SP),R2-      :SET-R2=-ADDRESS-
671 003000 016667 000002 000000G-      MOV-      2(SP),PREADD-      :SET-UP-START-ADDRESS-
672 003006 005102-      COM-      R2-      :GET-ADDRESS-COMPLEMENT-
673 003010 010267 000000G-      MOV-      R2,CKDATA-      :SET-TEST-PATTERN=-ADDR.COM-
674 003014 004767 000224      JSR-      PC,WRITET-      :WRITE-CKDATA-TO-BYTE-TRANS-
675 003020 016702 000000G-      MOV-      PREADD,R2-      :SET-R2=-NEXT-ADDRESS-
676 003024 026667 000004 000000G-      CMP-      4(SP),PREADD-      :FINISHED-
677 003032 103365      BHIS-      1$      :NO-CONTINUE-
678      :
679 003034 016602 000002      MOV-      2(SP),R2-      :SET-R2=-ADDRESS-
680 003040 016667 000002 000000G-      MOV-      2(SP),PREADD-      :RESET-START-ADDRESS-
681 003046 005102-      COM-      R2-      :GET-ADDRESS-COMPLEMENT-
682 003050 010267 000000G-      MOV-      R2,CKDATA-      :SET-TEST-PATTERN=-ADDR.COM-
683 003054 004767 000224      JSR-      PC,READT-      :READ-AND-COMPARE-
684 003060 016702 000000G-      MOV-      PREADD,R2-      :SET-R2=-NEXT-ADDRESS-
685 003064 026667 000004 000000G-      CMP-      4(SP),PREADD-      :FINISHED-
686 003072 103365      BHIS-      2$      :NO-
687 003074 000207      RTS-      PC-
688      :
689      :
690      :      TEST-12-
691      :      ADDRESSING-TEST-
692      :
693      :
694 003076      T12TD::
695 003076 012767 000000 176376      MOV-      #0,TD$SW-      :RESET-TERM-DETECTOR-
696 003104 016667 000002 000000G-      MOV-      2(SP),PREADD-      :GET-START-ADDRESS-
697 003112 016767 000000G 000000G-      MOV-      CK2,CKDATA-      :GET-READ-PATTERN-
698 003120 004767 000160      JSR-      PC,READT-      :READ-AND-COMPARE-
699 003124 162767 000001 000000G-      SUB-      #1,PREADD-      :WRITE-AT-SAME-ADDRESS-
700 003132 016767 000000G 000000G-      MOV-      CK3,CKDATA-      :GET-WRITE-PATTERN-
701 003140 004767 000100      JSR-      PC,WRITET-
702 003144 026667 000004 000000G-      CMP-      4(SP),PREADD-
703 003152 103357      BHIS-      1$
704 003154 000207      RTS-      PC-
705      :
706 003156      T12TU::
707 003156 012767 000000 176376      MOV-      #0,TD$SW-      :RESET-TERM-DETECTOR-
```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

708 003164 016667 000004 000000G. MOV. 4(SP),PREADD. ;START ADDRESS = UPPER LIMIT
709 003172 016767 000000G-000000G-1$: MOV. CK2,CKDATA. ;READ PATTERN
710 003200 004767 000100 JSR. PC,READT. ;READ AND COMPARE
711 003204 162767 000001 000000G. SUB. #1,PREADD. ;BACK UP TO SAME ADDRESS
712 003212 016767 000000G-000000G. MOV. CK3,CKDATA. ;WRITE PATTERN
713 003220 004767 000020 JSR. PC,WRITET
714 003224 162767 000002 000000G. SUB. #2,PREADD. ;BACK UP 1
715 003232 026667 000002 000000G. CMP. 2(SP),PREADD.
716 003240 003754 BLE. 1$
717 003242 000207 RTS. PC
718
719
720
721
722
723
724 003244 WRITET:
725 003244 012767 000030 176362. MOV. #T$BTAR,TD$TAW ;SELECT BYTE TRANS REG
726 003252 016767 000000G-176364. MOV. PREADD,TD$TDW ;MOVE ADDR TO TRANSFER REG
727 003260 012767 000020 176362. MOV. #T$BT,TD$TAW ;SELECT BYTE TRANS MEMORY
728 003266 016767 000000G-176364. MOV. CKDATA,TD$TDW ;WRITE BYTE TRANS
729 003274 062767 000001 000000G. ADD. #1,PREADD. ;BUMP ADDRESS
730 003302 000207 RTS. PC
731
732
733
734
735
736 003304 READT:
737 003304 012767 000030 176362. MOV. #T$BTAR,TD$TAW ;SELECT BYTE TRANS REG
738 003312 016767 000000G-176364. MOV. PREADD,TD$TDW ;MOVE ADDR TO TRANSFER REG
739 003320 012767 000020 176372. MOV. #T$BT,TD$TAR ;SELECT BYTE TRANS MEMORY
740 003326 016701 176374. MOV. TD$TDR,R1 ;READ BYTE TRANS
741 003332 026701 000000G. CMP. CKDATA,R1 ;CHECK AGAINST PATTERN
742 003336 001402. BEQ. +6 ;OK, CONTINUE
743 003340 004767 000010 JSR. PC,PRPERT ;PRINT ERROR MESSAGE
744 003344 062767 000001 000000G. ADD. #1,PREADD. ;BUMP ADDRESS
745 003352 000207 RTS. PC
746
747
748
749
750
751 003354 PRPERT:
752 003354 010146 SAVE. R1,R2,R3
753 003354 010246 MOV. R1,-(SP)
754 003356 010346 MOV. R2,-(SP)
755 003360 010346 MOV. R3,-(SP)
756
757
758
759
760
761
762 003362 016767 000000G-000000G. MOV. PREADD,ERRADD.
763 003370 012767 000001 000000G. MOV. #1,ERRCT. ;PRINT ONE WORD
764 003376 012767 000030 176362. MOV. #T$BTAR,TD$TAW ;SET UP READ ADDRESS
765 003404 016767 000000G-176364. MOV. PREADD,TD$TDW ;ADDRESS TO READ
766 003412 012767 000020 176372. MOV. #T$BT,TD$TAR ;SELECT TRANSFER REG
767 003420 016767 176374 000000G. MOV. TD$TDR,ERW1 ;READ OUTPUT BUFFER
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```
762.
763 003426 012702 000005      ;
764 003432 012767 000030 176362 1$: MOV. #5,R2. ;NUMBER OF RETRIES
765 003440 016767 000000G 176364 MOV. #T$BTAR,TD$TAW ;SET UP READ ADDRESS
766 003446 012767 000020 176372 MOV. PREADD,TD$TDW. ;ADDRESS TO READ
767 003454 016703 176374 MOV. #T$BT,TD$TAR. ;SELECT BYTE TRANSLATOR
768 003460 020301 MOV. TD$TDR,R3 ;READ OUTPUT BUFFER
769 003462 001003 CMP. R3,R1
770 003464 005302 BNE. 2$ ;NOT THE SAME ERROR
771 003466 001361 DEC. R2.
772 003470 000403 BNE. 1$
773 BR 3$ ;REPORT ERROR
774 003472 012767 177777 000000G 2$: MOV. #-1,ERRCT. ;INDICATE UNPREDICTABLE RESULTS
775 003500 004767 000000G 3$: JSR. PC,MEMERR.
776
777 003504 RESTOR. R1,R2,R3
      003504 MOV. (SP)+,R3
      003506 MOV. (SP)+,R2.
      003510 MOV. (SP)+,R1
778 003512 000207 RTS. PC
779
780 000001 ;.END.
```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
. ABS.      000000      000
          000000      001
SRCOFF.    000122.      002.
FDSCOF.    000010      003
FNOFFS.    000022.      004
TTEST.     003514      005
ERRORS DETECTED: 0
```

Approved For Release 2005/07/10 : CIA-RDP85-00514R000200020001-3

TMT.TSK:3 MEMORY-ALLOCATION-MAP: TKB: PAGE:1
27-MAR-80 Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

TASK NAME : TMT.
PARTITION NAME : GEN
IDENTIFICATION : 0736
TASK UIC : [7.5]
STACK LIMITS : 040212 041211 001000 00512.
PRG XFR ADDRESS : 045042.
TOTAL ADDRESS WINDOWS : 2.
TASK IMAGE SIZE : 6240. WORDS.
TASK ADDRESS LIMITS : 040000 070253
R-W DISK BLK LIMITS : 000002 000032 000031 00025.

*** ROOT SEGMENT: TMT.

R/W MEM LIMITS : 040000 070253 030254 12460.
DISK BLK LIMITS : 000002 000032 000031 00025.

MEMORY-ALLOCATION-SYNOPSIS:

SECTION...	TITLE...	IDENT...	FILE...
BLK: (RW, I, LCL, REL, CON)	041212 016562 07530.		
	041212 011506 04934.	TMT	TMT.OBJ:1
FDSCOF: (RW, I, LCL, ABS, CON)	000000 000000 00000.		
	000000 000000 00000.	TMT	TMT.OBJ:1
	000000 000000 00000.	TTEST	TTEST.OBJ:1
FNOFFS: (RW, I, LCL, ABS, CON)	000000 000000 00000.		
	000000 000000 00000.	TMT	TMT.OBJ:1
	000000 000000 00000.	TTEST	TTEST.OBJ:1
SRCOFF: (RW, I, LCL, ABS, CON)	000000 000000 00000.		
	000000 000000 00000.	TMT	TMT.OBJ:1
	000000 000000 00000.	TTEST	TTEST.OBJ:1
TTEST: (RW, I, LCL, REL, CON)	057774 003514 01860.		
	057774 003514 01860.	TTEST	TTEST.OBJ:1
\$\$FSR1: (RW, D, GBL, REL, OVR)	063510 001020 00520.		
	063510 001020 00520.	TMT	TMT.OBJ:1
\$\$FSR2: (RW, D, GBL, REL, CON)	064530 000104 00060.		
\$\$RESL: (RW, I, LCL, REL, CON)	064634 003420 01800.		
\$\$RESM: (RW, I, LCL, REL, CON)	132000 007656 04014.		

GLOBAL SYMBOLS:

CKDATA: 042506-R	ERW1	042522-R	STUFF	057774-R	T1I	061134-R	T12ID	061362-R	T12TU	063152-R	T7F	060206-R
CK2: 042510-R	ERW2	042524-R	STUFI	061046-R	T10	061750-R	T12IU	061446-R	T6F	060204-R	T7I	061240-R
CK3: 042512-R	ERW3	042526-R	STUFO	061674-R	T1T	062660-R	T12OD	062166-R	T6I	061236-R	T70	062056-R
ERRADD: 042516-R	MEMERR	047752-R	STUFT	062604-R	T12FD	060334-R	T12OU	062246-R	T60	062054-R	T7T	062762-R
ERRCT: 042520-R	PREADD	042514-R	T1F	060066-R	T12FU	060422-R	T12TD	063072-R	T6T	062760-R		

TMT:FSK:3 MEMORY-ALLOCATION-MAP: TKB:
TMT: 27-MAR-80

PAGE 2

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

*** TASK-BUILDER-STATISTICS:

TOTAL-WORK-FILE-REFERENCES: 11893.
WORK-FILE-READS: 0.
WORK-FILE-WRITES: 0.
SIZE-OF-CORE-POOL: 6634. WORDS (25. PAGES)
SIZE-OF-WORK-FILE: 2048. WORDS (8. PAGES)

ELAPSED-TIME:00:00:15

.TITLE: TMAIN.
.LIST: MEB.

TERM DETECTOR: 'MANUAL' DEBUGGING AIDS.
MAIN: MODULE.

THIS MODULE PASSES CONTROL TO ITS SUB-MODULES (COMMAND
MODULES) BASED ON INFORMATION IN THE COMMAND LINE (TERMINAL
INPUT). THE SUB-MODULES CONTINUE TO PARSE THE COMMAND LINE,
TRANSFERRING CONTROL TO THEIR SUB-ROUTINES. THE SUB-ROUTINES
DO THE ACTUAL INTERFACING WITH THE HARDWARE. THERE ARE
THREE LEVELS OF CONTROL. FOR EXAMPLE, TAKE THE COMMAND:
>LD:OB:0
THE 'LD' REPRESENTS THE FIRST LEVEL OF CONTROL AND IS PARSED
BY THE MODULE TMAIN. THE 'OB' REPRESENTS THE SECOND LEVEL OF
CONTROL AND IS PARSED BY THE TMAIN SUB-MODULE TMEM. THE THIRD
LEVEL OF CONTROL, A SUB-ROUTINE OF TLOAD, ACTUALLY CONTROLS
THE LOADING OF THE HARDWARE. TMAIN CONTAINS THE COMMANDS
'RS' (RESET) AND 'EX' (EXIT).

SUB-MODULES OF TMAIN:
TREG: REGISTER COMMANDS.
TMEM: MEMORY COMMANDS.
TDATA: INPUT BUFFER AND BYTE TRANSLATOR COMMANDS.
TRUN: RUN TD COMMANDS.

CONTROL IS RETURNED TO TMAIN WHEN A COMMAND HAS BEEN
EXECUTED OR ANND ERROR ENCOUNTERED.

TMAIN ALSO CONTAINS DATA AND SUBROUTINES COMMON TO ALL OF ITS
SUB-MODULES.

ASSEMBLY:
MCR>MAC TMAIN,LP=IM04,TMAIN: FROM: [5,3].

TASK BUILD:
1. TD STAND-ALONE PACK COMMAND FILE AIDTD.CMD.
AIDTD/DA,AIDTD=TMAIN,TREG,TDATA,TRUN.

PAR=PAR14K.
ASG=TT0:1:2.

2. NPIC SYSTEM COMMAND FILE AIDTD.CMD (USING OVERLAYS)
***** NOTE *****
TASK BUILD MUST BE DONE ON NPIC /04
MAPPING DOES NOT COME OUT RIGHT WHEN TKB IS DONE ON /45
EG. ADDRESS OF SYSTEM ROUTINE \$DIV COMES OUT INCORRECTLY.
***** NOTE *****

AIDTD,AIDTD=AIDTD/MP.
PAR=GEN:40000:40000

TMAIN: MACRO-M1110 27-MAR-80 13:39 PAGE 5-1

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
58      ; RESLIB=[1,1]FCSRES/RO.  
59      ; ASG=TT0:1:2.  
60      ; //  
61      ;  
62      ; OVERLAY:DESCRIPTION: AIDTD:ODL;  
63      ; .ROOT: TMAIN-*(TREG, TMEM, TDATA, TRUN)  
64      ; .END
```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

66      ;
67      ;
68      ; LOCAL DATA AREAS
69      ;
70      ;
71      ;
72      ; .MCALL QIOW$S,QIO$S,EXIT$S,ABRT$S,GCML$,GCMLB$,FSRSZ$,CLEF$S
73      ; .MCALL ASTX$S,FDBDF$,FDRC$A,FDOP$A,NMBLK$,OPEN$R,GET$,CLOSE$
74      ;
75      ; .GLOBL IO,WVB,IO,RVB,IO,ATA,IO,DET
76      ;
77      000001 EFN.1 == 1 ;EVENT FLAG FOR TERMINAL QIO'S
78      000001 LUN.TT == 1 ;TERMINAL LUN
79      000002 CMILUN == 2 ;GCML LUN
80      000003 TXTLUN == 3 ;LUN FOR DISK 'DATA' FILE
81      000004 TRLUN == 4 ;LUN FOR DISK BYTE TRANSLATOR OVERRIDE FILE
82      ;
83      000004 LOOP == 4 ;LOOP ON TEST FLAG
84      000010 ONCE == 10 ;PRINT ONE WORD ONLY
85      000020 ASTFLG == 20 ;QIO +AST ISSUED
86      000100 RIDE == 100 ;BYTE TRANSLATOR OVERRIDE IN EFFECT
87      000200 RP == 200 ;REPEAT PROMPT
88      000400 OP == 400 ;PRINT OUTPUT BUFFER CONTENTS
89      001000 ON == 1000 ;DO NOT PRINT OB CONTENTS
90      002000 OUT == 2000 ;CONTROL PRINTING OF MEMORY CONTENTS
91      ;
92      ;
93      ;
94      000000 003654 076640 MYSELF: .RAD50 /AIDTD/
95      000004 STAT: .BLKW 2
96      000010 ERWORD: .WORD 0 ;INDEX VALUE FOR ERROR MESSAGE TABLE
97      000012 000000 BINWD: .WORD 0 ;TARGET FOR NUMERIC CONVERSIONS FROM ASCII
98      000014 000000 BASE: .WORD 0 ;ALL PURPOSE FLAG
99      000016 000000 APLACE: .WORD 0 ;PRELIM BIT SETTINGS FOR TD CONTROL REG (TD$CTW)
100 000020 GCMBUF: .BLKW 41 ;COMMAND LINE BUFFER
101 000142 000000 GCMLN: .WORD 0 ;COMMAND LINE LENGTH
102 000144 000000 GCMPNT: .WORD 0 ;COMMAND LINE POINTER
103 000146 000000 ASTWRD: .WORD 0 ;RECEIVER FOR AST CHAR
104 000150 000000 RTNPT: .WORD 0 ;RTN ADDR SAVE AREA
105 000152 000000 MSTR1: .WORD 0 ;START ADDR FOR MEMORY LOADING/PRINTING
106 000154 000000 MSTR2: .WORD 0 ;WORKING ADDR FOR LOADING/PRINTING
107 000156 000000 MEND: .WORD 0 ;END ADDRESS FOR MEMORY LOADING/PRINTING
108 000160 000000 INCVAL: .WORD 0 ;MEMORY INCREMENT VALUE
109 000162 000000 RSPCNT: .WORD 0 ; (COMMAND LINE) RESPONSE COUNT
110 000164 000000 UPLIM: .WORD 0 ;MEMORY UPPER LIMIT
111 000166 000000 COUNT: .WORD 0 ;COUNTER FOR 8 TO 6-BIT CONVERSION
112 000170 000000 DATALN: .WORD 0 ;LENGTH OF 'DATA' BUFFER
113 000172 000000 DPLUS: .WORD 0 ;WORK COUNTER FOR 'DATA' RTN
114 000174 000000 OFF6: .WORD 0 ;OFFSET INTO 'DATA' BUFFER
115 000176 000000 CHLEN: .WORD 0 ;LENGTH OF CHANGE DATA + CHANGE START ADDRESS
116      ;
117 000200 WWORDS: ; CONVERTED NUMERIC VALUES FROM COMM LINE
118 000200 000000 DATA1: .WORD 0
119 000202 000000 DATA2: .WORD 0
120 000204 000000 DATA3: .WORD 0
121      ;
122      ; MEMORY LIMITS

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

123
124 000206 007777 AHIGH:: .WORD 7777 ;FSA-A
125 000210 000000 ALOW:: .WORD 0
126 000212 007777 BHIGH:: .WORD 7777 ;FSA-B
127 000214 000000 BLOW:: .WORD 0
128 000216 007777 CHIGH:: .WORD 7777 ;FSA-C
129 000220 000000 CLOW:: .WORD 0
130 000222 163776 IHIGH:: .WORD 163776 ;INPUT-BUFFER
131 000224 160000 ILOW:: .WORD 160000
132 000226 003776 OHIGH:: .WORD 3776 ;OUTPUT-BUFFER
133 000230 000000 OLOW:: .WORD 0
134 000232 000377 THIGH:: .WORD 255 ;BYTE-TRANSLATOR
135 000234 000000 TLOW:: .WORD 0
136
137
138 ;
139 ; ROUTINE-CONTROL-TABLES
140 ; COMMAND-MNEMONICS-AND-ASSOCIATED-ROUTINE-ADDRESSES
141 ;
142 ;
143 CTBL::
143 000236 123 124 .ASCII /ST/ ;STORE-INTO-A-REGISTER
144 000240 000000G .WORD ST1
145 000242 122 105 .ASCII /RE/ ;READ-A-REGISTER
146 000244 000000G .WORD RE1
147 000246 114 104 .ASCII /LD/ ;LOAD-A-MEMORY
148 000250 000000G .WORD LD1
149 000252 120 122 .ASCII /PR/ ;PRINT-MEMORY-CONTENTS-TO-CONSOLE
150 000254 000000G .WORD PR1
151 000256 104 101 .ASCII /DA/ ;SIMULATE-DATA-BASE
152 000260 000000G .WORD DA1
153 000262 103 110 .ASCII /CH/ ;CHANGE-SIMULATED-DATA
154 000264 000000G .WORD CH1
155 000266 114 111 .ASCII /LI/ ;LIST-SIMULATED-DATA
156 000270 000000G .WORD LI1
157 000272 105 116 .ASCII /EN/ ;CONTROL-LENGTH-OF-SIMULATED-DATA
158 000274 000000G .WORD EN1
159 000276 104 106 .ASCII /DF/ ;READ-SIMULATED-DATA-OR-BYTE-TRANSLATOR
160 000300 000000G .WORD DF1 ;DATA-IN-FROM-DISK
161 000302 124 122 .ASCII /TR/ ;TRANSFER-(BYTE-TRANS-OR-INPUT-BUFFER)
162 000304 000000G .WORD TR1 ;FROM-PROGRAM-MEMORY-TO-TD
163 000306 122 123 .ASCII /RS/ ;RESET-TERM-DETECTOR
164 000310 010606 .WORD RS1
165 000312 122 125 .ASCII /RU/ ;RUN-TERM-DETECTOR
166 000314 000000G .WORD RU1
167 000316 105 130 .ASCII /EX/ ;EXIT-TD-EXERCISER
168 000320 010620 .WORD EX1
169 000015 CNUM. == <.-CTBL>/4

```

```

171      ;
172      ;
173      ;      'DATA' TABLES.
174      ;      DTBL.  =.      6-BIT TABLE.
175      ;      DSAVE. =.      ASCII MIRROR OF 8-BIT INPUT.
176      ;
177      ;
178      DTBL:: .BLKW. <<1364.*3>>8.>+1
179      DSAVE:: .BLKB. 1364.
180      DSEND::
181      .BYTE. 0
182      .EVEN.
183      ;
184      ;      CONVERSION TABLE FOR 'DATA'
185      ;
186      TR6TBL::
187      .BYTE. -1,-1,-1,-1,-1      ;CODES 0 - 11 UNUSED
188      .BYTE. -1,-1,-1,-1,-1
189      .BYTE. 75      ;LINE FEED
190      .BYTE. -1,-1      ;CODES 13 -14 UNUSED
191      .BYTE. 74      ;CARRIAGE RETURN
192      .BYTE. -1,-1,-1,-1,-1,-1      ;CODES 16 -37 UNUSED
193      .BYTE. -1,-1,-1,-1,-1,-1
194      .BYTE. -1,-1,-1,-1,-1,-1
195      .BYTE. 40
196      .BYTE. 41      ;SPACE
197      .BYTE. 42      ;EXCLAMATION POINT
198      .BYTE. 43      ;QUOTES
199      .BYTE. 44      ;POUND SIGN
200      .BYTE. 45      ;DOLLAR
201      .BYTE. 46      ;PERCENT
202      .BYTE. 47      ;AMPERSAND
203      .BYTE. 50      ;APOSTROPHE
204      .BYTE. 51      ;LEFT PAREN
205      .BYTE. 52      ;RIGHT PAREN
206      .BYTE. 53      ;ASTERISK
207      .BYTE. 54      ;PLUS
208      .BYTE. 55      ;COMMA
209      .BYTE. 56      ;HYPHEN
210      .BYTE. 57      ;PERIOD
211      .BYTE. 60      ;SLASH
212      .BYTE. 61      ;0
213      .BYTE. 62      ;1
214      .BYTE. 63      ;2
215      .BYTE. 64      ;3
216      .BYTE. 65      ;4
217      .BYTE. 66      ;5
218      .BYTE. 67      ;6
219      .BYTE. 70      ;7
220      .BYTE. 71      ;8
221      .BYTE. 72      ;9
222      .BYTE. 73      ;COLON
223      .BYTE. -1      ;SEMI-COLON
224      .BYTE. -1      ;LEFT ANGLE BRACKET NOT USED
225      .BYTE. -1      ;EQUAL SIGN NOT USED
226      .BYTE. 77      ;RIGHT ANGLE BRACKET NOT USED
227      .BYTE. 0      ;QUESTION MARK
                     ;USE AT SIGN FOR DOCUMENT START

```

228 005151	001	.BYTE	01	:A
229 005152	002	.BYTE	02	:B
230 005153	003	.BYTE	03	:C
231 005154	004	.BYTE	04	:D
232 005155	005	.BYTE	05	:E
233 005156	006	.BYTE	06	:F
234 005157	007	.BYTE	07	:G
235 005160	010	.BYTE	10	:H
236 005161	011	.BYTE	11	:I
237 005162	012	.BYTE	12	:J
238 005163	013	.BYTE	13	:K
239 005164	014	.BYTE	14	:L
240 005165	015	.BYTE	15	:M
241 005166	016	.BYTE	16	:N
242 005167	017	.BYTE	17	:O
243 005170	020	.BYTE	20	:P
244 005171	021	.BYTE	21	:Q
245 005172	022	.BYTE	22	:R
246 005173	023	.BYTE	23	:S
247 005174	024	.BYTE	24	:T
248 005175	025	.BYTE	25	:U
249 005176	026	.BYTE	26	:V
250 005177	027	.BYTE	27	:W
251 005200	030	.BYTE	30	:X
252 005201	031	.BYTE	31	:Y
253 005202	032	.BYTE	32	:Z
254 005203	033	.BYTE	33	:LEFT SQ BRACKET - ZONE
255 005204	034	.BYTE	34	:BACK SLASH - SUBZONE
256 005205	035	.BYTE	35	:RGT SQ BRACKET - PARA
257 005206	036	.BYTE	36	:UP ARROW - SENT
258 005207	037	.BYTE	37	:UNDERBAR - CLOSE UP
259				:CODES 140 - 177 NOT USED
260		.NLIST	MEB	
261	000040	.REPT	40	
262		.BYTE	-1	
263		.ENDR		
264		.LIST	MEB	
265		.EVEN		

TMAIN: M 000 M1110 27-MAR-80 13:39 PAGE 8

```

267      :
268      :
269      :      DEFAULT BYTE TRANSLATOR TABLE
270      :
271      :
272      BTRANS:
273      005250 002077      .WORD 002077      ; DOCUMENT START
274      005252 000001      .WORD 001        ; A
275      005254 000002      .WORD 002        ; B
276      005256 000003      .WORD 003        ; C
277      005260 000004      .WORD 004        ; D
278      005262 000005      .WORD 005        ; E
279      005264 000006      .WORD 006        ; F
280      005266 000007      .WORD 007        ; G
281      005270 000010      .WORD 010        ; H
282      005272 000011      .WORD 011        ; I
283      005274 000012      .WORD 012        ; J
284      005276 000013      .WORD 013        ; K
285      005300 000014      .WORD 014        ; L
286      005302 000015      .WORD 015        ; M
287      005304 000016      .WORD 016        ; N
288      005306 000033      .WORD 033        ; O
289      005310 000034      .WORD 034        ; P
290      005312 000021      .WORD 021        ; Q
291      005314 000022      .WORD 022        ; R
292      005316 000023      .WORD 023        ; S
293      005320 000024      .WORD 024        ; T
294      005322 000025      .WORD 025        ; U
295      005324 000026      .WORD 026        ; V
296      005326 000027      .WORD 027        ; W
297      005330 000030      .WORD 030        ; X
298      005332 000031      .WORD 031        ; Y
299      005334 000032      .WORD 032        ; Z
300      005336 040033      .WORD 040033      ; ZONE MARK
301      005340 100034      .WORD 100034      ; SUBZONE MARK
302      005342 010035      .WORD 010035      ; PARAGRAPH MARK
303      005344 004036      .WORD 004036      ; SENTENCE
304      005346 000437      .WORD 437        ; CLOSE UP
305      005350 000400      .WORD BIT8      ; SPACE
306      005352 000400      .WORD BIT8      ; EXCL POINT
307      005354 000400      .WORD BIT8      ; QUOTE
308      005356 000301      .WORD 301        ; POUND
309      005360 000302      .WORD 302        ; DOLLAR
310      005362 000400      .WORD BIT8      ; PERCENT
311      005364 000400      .WORD BIT8      ; AMPERSAND
312      005366 000400      .WORD BIT8      ; APOSTROPHE
313      005370 000400      .WORD BIT8      ; LEFT PAREN
314      005372 000400      .WORD BIT8      ; RIGHT PAREN
315      005374 000400      .WORD BIT8      ; ASTERISK
316      005376 000400      .WORD BIT8      ; PLUS
317      005400 000400      .WORD BIT8      ; COMMA
318      005402 000400      .WORD BIT8      ; HYPHEN
319      005404 000400      .WORD BIT8      ; PERIOD
320      005406 000400      .WORD BIT8      ; SLASH
321      005410 000257      .WORD 257        ; 0
322      005412 000277      .WORD 277        ; 1
323      005414 000057      .WORD 057        ; 2

```


Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

324 005416 000077	.WORD 077	:3	63
325 005426 000117	.WORD 117	:4	64
326 005422 000137	.WORD 137	:5	65
327 005424 000157	.WORD 157	:6	66
328 005426 000177	.WORD 177	:7	67
329 005430 000217	.WORD 217	:8	70
330 005432 000237	.WORD 237	:9	71
331 005434 000400	.WORD BIT8	:COLON	72
332 005436 000400	.WORD BIT8	:SEMI-COLON	73
333 005440 000400	.WORD BIT8	:	74
334 005442 000400	.WORD BIT8	:	75
335 005444 000303	.WORD 303	:	76
336 005446 000400	.WORD BIT8	:DECIMAL	77
337 005450 000	BTEND: .BYTE 0		
338	.EVEN		
339	:		
340	: BYTE TRANSLATOR OVERRIDE TABLE		
341	:		
342 005452	BTOVER: .BLKW <BTEND-BTRANS>/2		
343 005452			

```

345      ;
346      ;
347      ; PRINT LINE
348      ;
349 005652 015 012
350 005654
351      ;
352      000116
353      ;
354      ;
355      ;
356      ;
357      ;
358      ;
359      ;
360      ;
361 005772 000
362 005773 015 012 015
363 005777 124 105 123
364 006015 015 012
365 006017 105 116 104
366 006043 015 012
367 006045 105 116 104
368 006073 015 012
369 006075 105 116 124
370 006136 015 012 015
371 006144 124 105 122
372 006207 015 012 000
373 006212 015 012
374 006214 111 114 114
375 006240 015 012
376 006242 106 117 122
377 006317 015 012
378 006321 111 114 114
379 006355 015 012
380 006357 123 124 101
381 006417 015 012
382 006421 123 124 101
383 006460 015 012
384 006462 111 116 126
385 006505 015 012
386 006507 102 131 124
387 006546 015 012
388 006550 105 115 120
389 006621 015 012
390 006623 105 122 122
391 006671 015 012
392 006673 105 122 122
393 006741 015 012
394 006743 105 122 122
395 007000 015 012
396 007002 105 122 122
397 007037 015 012
398 007041 111 116 126
399 007067 015 012
400 007071 047 104 101
401 007132 015 012

```

PRINT:: .BYTE 15,12 ; PRECEDE PRINT LINE WITH CRLF

.NLIST MEB
.REPT 78
.BYTE 40
.ENDR
.LIST MEB

TABLE OF MESSAGES

.BYTE 0
.BYTE 15,12,15,12
.ASCIZ /TEST(S) ENDED/
.BYTE 15,12
.ASCIZ /END OF FILE REACHED/
.BYTE 15,12
.ASCIZ /END OF MEMORY REACHED/
.BYTE 15,12
.ASCIZ /ENTER ANY CHARACTER TO EXIT LOOP/
.BYTE 15,12,15,12,15,12
.ASCIZ /TERM DETECTOR MANUAL DEBUGGING AIDS/
.BYTE 15,12,0
.BYTE 15,12
.ASCIZ /ILLEGAL ODD ADDRESS/
.BYTE 15,12
.ASCIZ /FOR BT. 'CHANGE' COMMAND MUST PRECEED 'LIST'/
.BYTE 15,12
.ASCIZ /ILLEGAL CHARACTER IN 'DATA'/
.BYTE 15,12
.ASCIZ /START ADDRESS OUT OF RANGE HIGH/
.BYTE 15,12
.ASCIZ /START ADDRESS OUT OF RANGE LOW/
.BYTE 15,12
.ASCIZ /INVALID TABLE NAME/
.BYTE 15,12
.ASCIZ /BYTE TRANSLATOR TABLE OVERFLOW/
.BYTE 15,12
.ASCIZ /EMPTY INPUT LINE IN BYTE TRANSLATOR FILE/
.BYTE 15,12
.ASCIZ /ERROR ON READ OF BYTE TRANSLATOR FILE/
.BYTE 15,12
.ASCIZ /ERROR ON OPEN OF BYTE TRANSLATOR FILE/
.BYTE 15,12
.ASCIZ /ERROR ON READ OF 'DATA' FILE/
.BYTE 15,12
.ASCIZ /ERROR ON OPEN OF 'DATA' FILE/
.BYTE 15,12
.ASCIZ /INVALID FILE MNEMONIC/
.BYTE 15,12
.ASCIZ /'DATA' COMMAND MUST PRECEED 'END'/
.BYTE 15,12

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
010142 000120 .WORD 80.
440 010262 FDOP$A TXTLUN,,TXTNMB
010164 003 .BYTE TXTLUN.
010170 010262 .WORD TXTNMB.
441 010262 TXTNMB: NMBLK$ DATA,TXT.
010270 014474 003100 .RAD50 /DATA/.
010274 000000 .WORD 0
010276 100324 .RAD50 /TXT/.
442. ;
443. ; FDB.FOR.BYTE TRANSLATOR.OVERRIDE
444. ;
445 010320 TRFDB: FDBDF$
446 010460 FDRC$A .GCMBUF,80.
010342 000020 .WORD GCMBUF.
010340 000120 .WORD 80.
447 010460 FDOP$A TXLUN,,TRNMB.
010362 004 .BYTE TXLUN.
010366 010460 .WORD TRNMB.
448 010460 TRNMB: NMBLK$ TRANS,TXT.
010466 077721 055170 .RAD50 /TRANS/.
010472 000000 .WORD 0
010474 100324 .RAD50 /TXT/.
449 010516 FSRSZ$ 2.
```

```

451      ;
452      ;
453      ; ENTER HERE
454      ;
455      ;
456 010516      START:
457 010516 004767 003052      JSR      PC,OUT1      ; ISSUE INFORMATION MESSAGE
458      ;
459      ;
460      ;
461      ; TOP OF COMMAND LOOP
462      ;
463      ;
464 010522      COMXX:
465 010522 004767 003212      JSR      PC,SELTST      ; PROMPT FOR COMMAND
466 010526 004767 002346      JSR      PC,FIND      ; FIND THE COMMAND
467 010532 103003      BCC      1$      ; OK, CONTINUE
468 010534 004767 003164      JSR      PC,ERR2      ; INVALID COMMAND
469 010540 000770      BR      COMXX
470 010542 022700 000002      1$: CMP      #2,R0      ; COMMANDS ARE 2 CHARS
471 010546 001403      BEQ      2$
472 010550 004767 003150      JSR      PC,ERR2      ; INVALID COMMAND
473 010554 000762      BR      COMXX      ; TRY AGAIN
474      ;
475 010556 012700 000015      2$: MOV      #CNUM,R0      ; R0 = NUMBER OF COMMANDS
476 010562 012702 000236      MOV      #CTBL,R2      ; R2 -> TABLE OF VALID COMMANDS
477 01056E 004767 002214      JSR      PC,SCAN      ; FIND MATCH IN TABLE
478 010572 103003      BCC      3$      ; OK, CONTINUE
479 010574 004767 003124      JSR      PC,ERR2      ; COMMAND NOT IN TABLE
480 010600 000750      BR      COMXX      ; TRY AGAIN
481      ;
482      ;
483      ; JUMP TO THE ROUTINE THAT GOVERNS THE COMMAND
484 010602 000171 000000      3$: JMP      @ (R1)

```

```
486      ;
487      ;
488      ; RESET
489      ;
490      ;
491 010606 RS1:
492 010606 012767 000000 176376 MOV. #0.TD$SW. :RESET.TD.
493 010614 000167 177702 JMP. COMXX.
494      ;
495      ;
496      ;
497      ;
498      ; EXIT
499      ;
500 010620 EX1:
501 010620 EXIT$S.
      010620 012746 MOV. (PC)+,-(SP)
      010622 063 .BYTE 51..1
      010624 104377 EMT. +0<377>
```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

503      ;
504      ;
505      ;
506      ;
507      ;
508      ;
509      ;
510      ;
511      ;
512      ;
513      ;
514      ;
515      ;
516      ;
517      ;
518      ;
519      ;
520      ;
521      ;
522      ;
523      ;
524      ;
525      ;
526      ;
527      ;
528      ;
529      ;
530      ;
531      ;
532      ;
533      ;
534      ;
535      ;
536      ;
537      ;
538      ;
539      ;
540      ;
541      ;
542      ;
543      ;
544      ;
545      ;
546      ;
547      ;
548      ;
549      ;
550      ;
551      ;
552      ;
553      ;
554      ;
555      ;
556      ;
557      ;
558      ;
559      ;

```

SET-UP BEFORE PROMPTING ON ANY 'LD' COMMAND OR BEFORE EXECUTING ANY 'PR' COMMAND FOR FSA A, B, C. THE FIELDS SET-UP HERE ARE USED IN THE TMAIN PROMPTING SUBROUTINE. PDATA (LDCNTL FOR FSA'S) AND THE PRINT ROUTINE PDATA (PRCNTL FOR FSA'S).

INPUT:

2(SP) LOAD/PRINT START ADDRESS

4(SP) LOWER MEMORY LIMIT

6(SP) UPPER MEMORY LIMIT

APLACE - FSA SELECTION

OUTPUT:

INCVAL - VALUE BY WHICH MEMORY ADDRESS INCREMENTS

MEND - ADDRESS OF LAST WORD IN MEMORY TO BE LOADED/PRINTED

RSPCNT - NUMBER OF WORDS TO EXPECT IN COMMAND LINE (MEMORY WIDTH)

C-BIT CLEAR NO ERROR (START ADDRESS IS IN RANGE)

C-BIT SET ERROR IN RANGE

MEMORY	INCVAL	RSPCNT
FSA A	1	3
FSA B	1	3
FSA C	1	3

532. 010626 BUFFSA:

533 010626 012767 000000 176376 MOV. #0, TD\$SW. ; RESET TD

534 010634 052767 020000 167154 BIS. #T\$SCLK, APLACE ; START CLOCK

535 010642 016767 167150 176360 MOV. APLACE, TD\$CTW. ; SET TD CONTROL REG

IF THE FIELD MEND IS NOT --1, THEN A 'LD' FOR 'PR' ROUTINE HAS PUT A MEMORY UPPER LIMIT IN THIS FIELD. EG. A COMMAND SUCH AS:

>LD AM 0 10

WOULD PUT 10 IN THE FIELD MEND.

543 010650 022767 177777 167300 CMP. #-1, MEND. ; MEMORY END ADDR SET UP

544 010656 001003 BNE. 10\$; YES

545 010660 016667 000004 167270 MOV. 4(SP), MEND. ; SET END ADDR = MEMORY UPPER LIMIT

546 010666 026766 167260 000002 10\$: CMP. MSTRT, 2(SP) ; IS LOAD ADDRESS IN RANGE (LOW)

547 010674 103003 BHIS. 1\$; OK, CONTINUE

548 010676 004767 002716 JSR. PC, ERR19 ; OUT OF RANGE

549 010702 000432 BR. BUFCX

550 010704 026766 167246 000004 1\$: CMP. MEND, 4(SP) ; IS ADDR IN RANGE (HIGH)

551 010712 101403 BLOS. 2\$; YES, CONTINUE

552 010714 004767 002674 JSR. PC, ERR20 ; OUT OF RANGE

553 010720 000423 BR. BUFCX

554 010722 026767 167224 167226 2\$: CMP. MSTRT, MEND. ; IS START ADDR LOWER THAN END ADDR

555 010730 101403 BLOS. 3\$; YES, CONTINUE

556 010732 004767 002656 JSR. PC, ERR20 ; OUT OF RANGE

557 010736 000414 BR. BUFCX ; EXIT

558

559 010740 012767 000001 167212 3\$: MOV. #1, INCVAL. ; BUFFERS INC BY 1

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

TMAIN: M 30-M1110 27-MAR-80 13:39 PAGE 12-1

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

560	010746	012767	000003	167206	MOV	#3, RSPCNT	: NUMBER OF SEGMENTS (FOR PROMPT)
561							
562	010754	011666	000004		MOV	(SP), 4(SP)	: MOVE RETURN ADDRESS
563	010760	062706	000004		ADD	#4, SP	: ADJUST SP (FOR MEM LIMITS)
564	010764	000241			CLC		
565	010766	000405			BR	BUFXX	: RETURN
566	010770	011666	000004	BUFCX:	MOV	(SP), 4(SP)	
567	010774	062706	000004		ADD	#4, SP	
568	011000	000261			SEC		
569	011002	000207		BUFXX:	RTS	PC	

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

571      ;
572      ;
573      ;
574      ;
575      ;
576      ;
577      ;
578      ;
579      ;
580      ;
581      ;
582      ;
583      ;
584      ;
585      ;
586      ;
587      ;
588      ;
589      ;
590      ;
591      ;
592      ;
593      ;
594      ;
595      ;
596      ;
597      ;
598      ;
599 011004 012767 000002 167146 BUFS2:: MOV. #2, INCVAL.      ;BUFFERS INC BY 2.
600 011012 000403          BR. SET.
601 011014 012767 000001 167136 BUFSET:: MOV. #1, INCVAL.      ;BUFFERS INC BY 1
602      ;
603 011022 012767 000000 176376 SET:  MOV. #0, TD$SJ.      ;RESET TD.
604 011030 016767 166762 176360 MOV.  APLACE, TD$CTW.    ;SET TD CONTROL REG.
605      ;
606      ;
607      ;
608      ;
609      ;
610      ;
611      ;
612 011036 022767 177777 167112 CMP.  #-1, MEND.      ;MEMORY END ADDR SET UP.
613 011044 001003          BNE. 10$.      ;YES.
614 011046 016667 000004 167102 MOV.  4(SP), MEND.    ;SET END ADDR = MEMORY UPPER LIMIT.
615 011054 026766 167072 000002 10$: CMP.  M$TRT, 2(SP)    ;IS LOAD ADDRESS IN RANGE (LOW).
616 011062 103003          BHIS. 1$.      ;OK, CONTINUE.
617 011064 004767 002530 JSR.  PC, ERR19    ;OUT OF RANGE.
618 011070 000446 BR.  BUFCX2.
619 011072 026766 167060 000004 1$: CMP.  MEND, 4(SP)      ;IS ADDR IN RANGE (HIGH)
620 011100 101403          BLJS. 2$.      ;YES, CONTINUE.
621 011102 004767 002506 JSR.  PC, ERR20    ;OUT OF RANGE.
622 011106 000437 BR.  BUFCX2.
623 011110 026767 167036 167040 2$: CMP.  M$TRT, MEND.    ;IS START ADDR LOWER THAN END ADDR.
624 011116 101403          BLOS. 3$.      ;YES, CONTINUE.
625 011120 004767 002470 JSR.  PC, ERR20    ;OUT OF RANGE.
626 011124 000430 BR.  BUFCX2.      ;EXIT.
627      ;

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

628	011126	032767	000001	167024	3\$:	BIT	#1, INCVAL	: IS MEMORY ADDR INCREMENT EVEN
629	011134	001013				BNE	5\$: NO, ANY START ADDR IS OK
630	011136	032767	000001	167006		BIT	#1, MSTRT	: IS START ADDRESS EVEN
631	011144	001004				BNE	4\$: NO, ERROR
632	011146	032767	000001	167002		BIT	#1, MEND	: IS END ADDRESS EVEN
633	011154	001403				BEQ	5\$: YES, CONTINUE
634	011156	004767	002416		4\$:	JSR	PC, ERR23	
635	011162	000411				BR	BUFCX2	
636								
637	011164	012767	000001	166770	5\$:	MOV	#1, RSPCNT	: NUMBER OF SEGMENTS (FOR PROMPT)
638	011172	011666	000004			MOV	(SP), 4(SP)	: MOVE RETURN ADDRESS
639	011176	062706	000004			ADD	#4, SP	: ADJUST SP (FOR MEM LIMITS)
640	011202	000241				CLC		
641	011204	000405				BR	BUFCX2	: RETURN
642	011206	011666	000004		BUFCX2:	MOV	(SP), 4(SP)	
643	011212	062706	000004			ADD	#4, SP	
644	011216	000261				SEC		
645	011220	000207			BUFCX2:	RTS	PC	

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

647      ;
648      ;
649      ;
650      ;
651      ;
652      ;
653      ;
654      ;
655      ;
656      ;
657      ;
658      ;
659      ;
660 011222      ;
661 011222 032767 000200 166564      ;
662 011230 001424      ;
663 011232 004767 000144      ;
664 011236 103460      ;
665 011240 102457      ;
666 011242 016767 166706 176362      ;
667 011250 016767 166724 176364      ;
668 011256 016767 166720 176364      ;
669 011264 016767 166714 176364      ;
670 011272 066767 166662 166654      ;
671 011300 000754      ;
672      ;
673 011302 004767 000074      ;
674 011306 103434      ;
675 011310 102433      ;
676 011312 016767 166636 176362      ;
677 011320 016767 166654 176364      ;
678 011326 016767 166650 176364      ;
679 011334 016767 166644 176364      ;
680 011342 066767 166612 166604      ;
681 011350 026767 166600 166600      ;
682 011356 101755      ;
683 011360 032767 000004 166426      ;
684 011366 001404      ;
685 011370 016767 166556 166556      ;
686 011376 000745      ;
687      ;
688 011400      ;
689 011400 000207      ;

```

LOAD CONTROL FOR FSA'S

INPUT:

INCVL - VALUE BY WHICH MEMORY ADDRESS INCREMENTS.

MSTR2 - MEMORY START ADDRESS.

MEND - MEMORY END ADDRESS.

DATA1, DATA2, DATA3 - SET-UP BY ROUTINE PDATA.

#RP-FLAG - DETERMINES WHETHER TO REPEAT PROMPT.

#LOOP-FLAG - DETERMINES WHETHER TO LOOP ON LOAD.

LDCNTL:

BIT #RP.BASE ; REPEAT PROMPT.

BEQ 2\$; NO. PROMPT ONCE.

1\$: JSR PC,PDATA ; READ DATA FROM COMMAND LINE

BCS CNTLX ; ERROR OR END OF MEMORY

BVS CNTLX ; <CR> RESPONSE TO PROMPT

MOV MSTR2,TD\$TAW ; MOVE ADDR TO TRANSFER REG.

MOV DATA1,TD\$TDW ; MOVE SEG 1

MOV DATA2,TD\$TDW ; MOVE SEG 2

MOV DATA3,TD\$TDW ; MOVE SEG 3

ADD INCVL,MSTR2 ; ADVANCE ADDRESS

BR 1\$; REPEAT

2\$: JSR PC,PDATA ; PROMPT ONCE

BCS CNTLX ; END OF MEMORY OR ERROR

BVS CNTLX ; <CR> RESPONSE TO PROMPT

3\$: MOV MSTR2,TD\$TAW ; SET UP TRANSFER ADDRESS

MOV DATA1,TD\$TDW ; MOVE SEG 1

MOV DATA2,TD\$TDW ; MOVE SEG 2

MOV DATA3,TD\$TDW ; MOVE SEG 3

ADD INCVL,MSTR2 ; ADVANCE ADDRESS

CMP MSTR2,MEND ; HAS UPPER MEMORY LIMIT BEEN REACHED

BLOS 3\$; NO. CONTINUE

BIT #LOOP.BASE ; LOOP ON ?

BEQ CNTLX ; NO. EXIT

MOV MSTR2,MSTR2 ; REINITIALIZE ADDRESS

BR 3\$

CNTLX: RTS PC

```

691      ;
692      ;
693      ;
694      ;
695      ;
696      ;
697      ;
698      ;
699      ;
700      ;
701      ;
702      ;
703      ;
704      ;
705      ;
706      ;
707      ;
708      ;
709      ;
710      ;
711      ;
712      ;
713      ;
714      ;
715      ;
716      ;
717      ;
718      ;
719      ;
720      ;
721      ;
722      ;
723      ;
724      ;
725      ;
726      ;
727      ;
728      ;
729      ;
730      ;
731      ;
732      ;
733      ;
734      ;
735      ;
736      ;
737      ;
738      ;
739      ;
740      ;
741      ;
742      ;
743      ;
744      ;
745      ;
746      ;
747      ;

```

PROMPTING CONTROL FOR LOADING ALL MEMORIES AND BUFFERS.
 READ NUMERIC DATA FROM THE COMMAND LINE, CONVERT AND STORE
 INTO A COMMON BUFFER, EG. IF THE INITIATING COMMAND IS:
 >LD-AM-0
 THIS ROUTINE WILL PUT OUT A PROMPT AND EXPECT 3 NUMERIC
 VALUES IN RETURN:
 >000000 000000 000000
 THIS ROUTINE CONVERTS THESE ASCII OCTAL VALUES INTO BINARY
 AND STORES THEM INTO DATA1, DATA2, DATA3

INPUT: (SET UP BY TMAIN SUBROUTINE BUFSET)
 MSTR2 - CURRENT MEMORY ADDRESS
 MEND - MEMORY UPPER ADDRESS LIMIT
 RSPCNT - NUMBER OF WORDS TO EXPECT IN COMMAND LINE

OUTPUT:
 WWORDS (DATA1, DATA2, DATA3) DEPENDING UPON RSPCNT.

C-BIT CLEAR, V-BIT CLEAR NORMAL RETURN
 C-BIT CLEAR, V-BIT SET <CR> RESPONSE TO PROMPT
 C-BIT SET, V-BIT CLEAR END OF MEMORY OR CONVERSION ERROR

REGISTERS 1, 4, 5 DESTROYED

PDATA::

```

CMP      MSTR2,MEND      ;UPPER MEMORY LIMIT REACHED
BLOS     10$             ;NO. CONTINUE
JSR      PC,ENDMEM       ;END OF MEMORY REACHED
BR       PDCX             ;SET CARRY AND EXIT
;
10$:     MOV      MSTR2,R1 ;PREPARE TO PRINT ADDRESS
MOV      #PRINT,R5        ;POINT TO PRINT LINE
JSR      PC,UNPK          ;CONVERT ADDRESSES
JSR      PC,CONSOL        ;PRINT OUT ADDRESS
;
MOV      #20040,GCMBLK+G:DPRM ;ERASE CR+LF
JSR      PC,GCONLY        ;PROMPT
MOV      #15,GCMBLK+G:DPRM ;RESTORE CR
MOV      #12,GCMBLK+G:DPRM+1 ;RESTORE LF
JSR      PC,FIND          ;LOCATE FIRST DATA WORD IN COMMAND LINE
BCS      PDVX             ;<CR> RESPONSE, EXIT
;
MOV      #WWORDS,R5       ;WORDS FROM COMMAND LINE GO INTO THIS TABLE
MOV      RSPCNT,R4         ;NUMBER OF WORDS TO EXPECT
JSR      PC,PACK          ;CONVERT WORD TO BINARY
BCC      3$               ;OK, CONTINUE
JSR      PC,ERR4          ;INVALID NUMERIC VALUE
BR       PDCX             ;NO. EXIT
;
3$:      MOV      BINWD,(R5)+ ;MOVE WORD TO TABLE
DEC      R4               ;SUB FROM LOOP COUNT
BEQ      PDCCX            ;FINISHED
JSR      PC,FIND          ;FIND NEXT WORD
BCC      4$               ;OK, CONTINUE
JSR      PC,ERR3          ;MISSING ADDRESS

```

748	011542	000416		BR	PDCX	
749			:			
750	011544	122711	000122	4\$:	CMPB	#1R, (R1)
751	011550	001355			BNE	2\$
752	011552	016725	166234		MOV	BINWD, (R5)+
753	011556	022704	000001		CMP	#1, R4
754	011562	001411			BEQ	PDCX
755	011564	016725	166222		MOV	BINWD, (R5)+
756	011570	000406			BR	PDCX
757			:			
758	011572	000241		PDVX:	CLC	
759	011574	000262			SEV	
760	011576	000405			BR	PDDX
761	011600	000242		PDCX:	CLV	
762	011602	000261			SEC	
763	011604	000402			BR	PDDX
764	011606	000241		PDCX:	CLC	
765	011610	000242			CLV	
766	011612	000207		PDDX:	RTS	PC

```

768      ;
769      ;
770      ; PRINT CONTROL FOR FSA'S.
771      ;
772      ; FIELDS ALREADY SET UP BY 'BUFFSA':
773      ; INPUT:
774      ; MSTR2 - MEMORY ADDRESS.
775      ; MEND - MEMORY END ADDRESS.
776      ; INCVAL - VALUE BY WHICH MEMORY ADDRESS INCREMENTS.
777      ; #LOOP FLAG - REPEAT MEMORY READ
778      ;
779      ; OUTPUT:
780      ; DATA1, DATA2, DATA3 (INPUT TO ROUTINE PRDATA)
781      ;
782      ;
783      ; PCNTL:
784      011614 016767 166334 176362 1$: MOV MSTR2,TD$TAW ;MOVE ADDR TO TRANSFER REG.
785      011622 016767 176374 166350 MOV TD$TDR,DATA1 ;MOVE SEG 1
786      011630 016767 176374 166344 MOV TD$TDR,DATA2 ;MOVE SEG 2
787      011636 016767 176374 166340 MOV TD$TDR,DATA3 ;MOVE SEG 3
788      011644 004767 000040 JSR PC,PRDATA ;PRINT MEMORY CONTENTS.
789      ;
790      011650 006767 166304 166276 ADD INCVAL,MSTR2 ;ADVANCE ADDRESS.
791      011656 006767 166272 166272 CMP MSTR2,MEND ;HAS UPPER MEMORY LIMIT BEEN REACHED
792      011664 101753 BLOS 1$ ;NO, CONTINUE.
793      011666 032767 000004 166120 BIT #LOOP,BASE ;LOOP
794      011674 001404 BEQ PCNTLX ;NO, EXIT.
795      011676 016767 166250 166250 MOV MSTR2,MSTR2 ;REINIT START ADDRESS.
796      011704 000743 BR 1$ ;AND REPEAT.
797      ;
798      011706 PCNTLX:
799      011706 000207 RTS PC

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

801      ;
802      ;
803      ;      COMMON PRINT ROUTINE.
804      ;      THIS ROUTINE CONVERTS VALUES IN THE FIELD 'WWORDS' TO
805      ;      ASCII/OCTAL (DEPENDING ON RSPCNT) AND PRINTS THEM (TT0)
806      ;
807      ;      INPUT:
808      ;      #OUT.FLAG - WHEN LOOP OPTION IS IN EFFECT, PRINT 1 WORD ONLY.
809      ;      #ONCE.FLAG - CONTROL PRINTING OF 1 WORD WHEN #OUT IS SET.
810      ;      RSPCNT - NUMBER OF WORDS TO PRINT (SEE 'BUFSET' RTH)
811      ;      MSTR2 - MEMORY ADDRESS (SEE 'BUFSET')
812      ;      WWORDS - MEMORY WORDS (DATA1, DATA2, DATA3, DATA4)
813      ;
814      ;      REGISTERS 1, 3, 4, 5 DESTROYED.
815      ;
816      ;
817 011710      PRDATA::
818 011710      032767 002000 166076      BIT #OUT.BASE ; OUTPUT CONTROL ON
819 011716      001407                      BEQ 10$      ; NO, SKIP OUTPUT CONTROL
820 011720      032767 000010 166066      BIT #ONCE.BASE ; ONE LINE PRINTED
821 011726      001026                      BNE PRDX      ; YES, EXIT
822 011730      052767 000010 166056      B13 #ONCE.BASE ; SET FLAG FOR NEXT TIME
823
824 011736      016701 166212      10$: MOV MSTR2,R1 ; FIRST CONVERT ADDRESS TO ASCII
825 011742      012705 005654      MOV #PRINT,R5 ; POINT TO PRINT LINE
826 011746      004767 001354      JSR PC,UNPK ; PERFORM CONVERSION
827
828 011752      012704 000200      MOV #WWORDS,R4 ; POINT TO WORDS FOR PRINTING
829 011756      016703 166200      MOV RSPCNT,R3 ; NUMBER OF MEM WORDS TO PRINT
830 011762      062705 000002      1$: ADD #2,R5 ; ADVANCE PRINT LINE POINTER
831 011766      012401      MOV (R4)+,R1 ; LOAD ONE WORD
832 011770      004767 001332      JSR PC,UNPK ; CONVERT IT TO ASCII
833 011774      005303      DEC R3 ; DEC WORD COUNT
834 011776      001371      BNE 1$ ; REPEAT
835 012000      004767 001366      JSR PC,CONSOL ; PRINT LINE
836
837 012004      000207      PRDX: RTS PC

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

879      ;
880      ;
881      ;
882      ;
883      ;
884      ;
885      ;
886      ;
887      ;
888      ;
889      ;
890      ;
891      ;
892      ;
893      ;
894      ;
895      ;
896      ;
897      ;
898      ;
899      ;
900      ;
901      ;
902      ;
903      ;
904      ;
905      ;
906      ;
907      ;
908      ;
909      ;
910      ;
911      ;
912      ;
913      ;
914      ;
915      ;
916      ;
917      ;
918      ;
919      ;
920      ;
921      ;
922      ;
923      ;
924      ;
925      ;
926      ;
927      ;
928      ;
929      ;
930      ;
931      ;
932      ;
933      ;
934      ;
935      ;

```

BUILD 6-BIT-BYTE 'DATA' BUFFER.
PROCESS ONE COMMAND LINE'S WORTH OF 8-BIT-BYTES.
CONVERT THEM TO 6-BIT-BYTES AND ADD THEM TO THE
6-BIT-BYTE TABLE. ALSO SAVE THE 8-BIT-BYTES IN
THEIR CURRENT FORM FOR USE BY 'LI' COMMAND.

INPUT:
GCMLN - LENGTH OF COMMAND LINE
COUNT - 6-BIT-BYTE POSITION (SEE ROUTINE 'OFFSET')
R1 - POINTER TO COMMAND LINE
R4 - POINTER TO 6-BIT-BYTE TABLE
R5 - POINTER TO 8-BIT-ASCII MIRROR IMAGE OF COMMAND
LINE INPUT TABLE

OUTPUT:
DPLUS - NUMBER OF 6-BIT-CHARACTERS ADDED TO 6-BIT-
BYTE TABLE IN CURRENT CALL OF DBLD.

DBLD::

MOV GCMLN,DPLUS ;NUMBER OF CHARS IN COMMAND LINE

THE STRINGS <12> AND <15> ARE ALLOWED IN THE COMMAND LINE
FOR THE ENTERING OF CARRIAGE RETURNS AND LINE FEEDS. A
CARRIAGE RETURN IS SAVED IN THE MIRROR TABLE AS A BINARY
15. A LINE FEED AS A BINARY 12. THE BRACKETS ARE RE-INSTATED
BY THE 'LI' COMMAND FOR PRINTING.

DBLD1: CMPB (R1),#< ;LOOK FOR <15> OR <12>
BNE NOCRLF ;NO ADJUSTMENT NEEDED
CMPB 3(R1),#< ;ARE BRACKETS CLOSED
BNE NOCRLF ;NO, TREAT BRACKET AS CHAR (INVALID)
CMPB 1(R1),#1 ;FIRST CHAR OF 12 OR 15
BEQ 1\$;OK, CONTINUE
JSR PC,ERR21 ;ILLEGAL 'DATA' CHARACTER
BR DBLDSX

PARSE THE SECOND OF THE 2 CHARS BETWEEN THE BRACKETS

1\$: MOV #10,R2 ;START WITH BINARY 10
CMPB 2(R1),#12 ;LINE FEED
BNE 2\$;NO, TRY CR
ADD #2,R2 ;MAKE R2 = 12
BR 4\$

2\$: CMPB 2(R1),#15 ;CARRIAGE RETURN
BEQ 3\$;OK, CONTINUE
JSR PC,ERR21 ;ILLEGAL 'DATA' CHARACTER
BR DBLDSX

MOVE BINARY VALUE TO MIRROR TABLE

3\$: ADD #5,R2 ;MAKE R2 = 15
4\$: MOVB R2,(R5)+ ;MOVE CR OR LF INTO MIRROR TABLE
ADD #4,R1 ;ADJUST COMMAND LINE POINTER
SUB #3,GCMLN ;ADJUST COMMAND LINE CHAR COUNT

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
936 012216 162767 000003 165746 SUB #3,DPLUS ;ADJUST 'DATA' CHAR COUNT FOR THIS LINE.
937 012224 000412 BR MOVTR
938 ;
939 ;
940 ;
941 012226 ;
942 012226 111125 NOCRLF: MOVB (R1),(R5)+ ;MOVE CHAR TO ASCII SAVE BUFFER.
943 012230 022705 005046 CMP #DSEND,R5 ;END OF BUFFER REACHED.
944 012234 002005 BGE 1$ ;NO, CONTINUE.
945 012236 004767 001322 JSR PC,ENDMEM ;ISSUE MESSAGE.
946 012242 105067 172600 CLRB DSEND ;CLEAR LAST BYTE OF TABLE.
947 012246 000512 BR DBLDSX ;AND EXIT.
948 012250 112102 1$: MOVB (R1)+,R2 ;LOAD CHAR FROM COMMAND LINE
949 ;
950 ;
951 ;
952 012252 ;
953 012252 116202 005050 MOVTR: MOVB TR6TBL(R2),R2 ;LOAD 6-BIT CHAR.
954 012256 002003 BGE 10$ ;VALID CHAR, CONTINUE.
955 012260 004767 001324 JSR PC,ERR21 ;ILLEGAL 'DATA' CHARACTER.
956 012264 000503 BR DBLDSX
957 ;
958 ;
959 ;
960 ;
961 ;
962 ;
963 ;
964 ;
965 ;
966 ;
967 ;
968 ;
969 012266 010203 10$: MOV R2,R3 ;COPY FOR LATER USE.
970 012270 005767 165672 TST COUNT ;POSITION = 0
971 012274 001004 BNE 1$ ;NO, TRY 1
972 012276 142714 000077 BICB #077,(R4) ;CLEAR AHEAD OF BIT SET.
973 012302 150314 BLSB R3,(R4) ;'OR' IN 6 BITS.
974 012304 000464 BR NEXTD ;GET NEXT CHAR.
975 ;
976 ;
977 ;
978 012306 122767 000001 165652 1$: CMPB #1,COUNT ;POSITION = 1
979 012314 001022 BNE 2$ ;NO, TRY 2.
980 012316 142703 000074 BICB #074,R3 ;CLEAR TOP 4 BITS OF 6-BIT CHAR (00111100)
981 012322 006303 ASL R3 ;SHIFT 2 BITS LEFT 6
982 012324 006303 ASL R3
983 012326 006303 ASL R3
984 012330 006303 ASL R3
985 012332 006303 ASL R3
986 012334 006303 ASL R3
987 012336 142714 000300 BICB #300,(R4) ;CLEAR AHEAD OF BIT SET.
988 012342 150324 BLSB R3,(R4)+ ;'OR' IN 6-BIT CHAR AND INCREMENT
989 ;
990 012344 010203 MOV R2,R3 ;LOAD ORIGINAL CHAR.
991 012346 006203 ASR R3 ;LEAVE 4 BITS OF 6-BIT CHAR.
992 012350 006203 ASR R3
```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

993 012352 142714 000017 BICB #017, (R4) ;CLEAR AHEAD OF BIT SET
994 012356 150314 BISB R3, (R4) ;'OR' IN 6-BIT CHAR
995 012360 000436 BR NEXTD ;GET NEXT COMMAND LINE CHAR
996 ;
997 ;
998 ;
999 012362 022767 000002 165576 2$: CMP #2, COUNT ;IS PASS COUNT = 2
1000 012370 001022 BNE 3$ ;NO, TRY 3
1001 012372 142703 000060 BICB #060, R3 ;CLEAR 2 BITS (00110000)
1002 012376 006303 ASL R3 ;SHIFT 4 REMAINING BITS LEFT 4
1003 012400 006303 ASL R3
1004 012402 006303 ASL R3
1005 012404 006303 ASL R3
1006 012406 142714 000360 BICB #360, (R4) ;CLEAR AHEAD OF BIT SET
1007 012412 150324 BISB R3, (R4) + ;'OR' IN PART OF A 6-BIT CHAR
1008 ;
1009 012414 010203 MOV R2, R3 ;LOAD ORIGINAL CHAR
1010 012416 006203 ASR R3 ;SHIFT 4 BITS RIGHT 4
1011 012420 006203 ASR R3
1012 012422 006203 ASR R3
1013 012424 006203 ASR R3
1014 012426 142714 000003 BICB #3, (R4) ;CLEAR AHEAD
1015 012432 150314 BISB R3, (R4) ;'OR' IN PART OF 6-BIT CHAR
1016 012434 000410 BR NEXTD
1017 ;
1018 ;
1019 ;
1020 012436 006303 3$: ASL R3 ;SHIFT 6 BITS LEFT 2
1021 012440 006303 ASL R3
1022 012442 142714 000374 BICB #374, (R4) ;CLEAR AHEAD
1023 012446 150324 BISB R3, (R4) + ;'OR' IN 6 BITS
1024 012450 012767 177777 165510 MOV #1, COUNT ;COUNT WILL BE RESET TO ZERO
1025 ;
1026 012456 005267 165504 NEXTD: INC COUNT ;BUMP 6-BIT BYTE COUNTER
1027 012462 005367 165454 DEC GCMLN ;DEC BYTE COUNTER
1028 012466 001406 BEQ DBLDCX
1029 012470 000167 177406 JMP DBLD1 ;PROCESS NEXT INPUT CHARACTER
1030 ;
1031 012474 005067 165470 DBLDSX: CLR DATALN ;AS THOUGH RTN WAS NOT ENTERED
1032 012500 000261 SEC
1033 012502 000401 BR DBLDXX ;AND EXIT
1034 012504 000241 DBLDCX: CLC
1035 012506 000207 DBLDXX: RTS PC

```

```

1037      ;
1038      ;
1039      ; GET COMMAND LINE FROM TERMINAL
1040      ;
1041      ;
1042      ; OUTPUT:
1043      ; GCMBUF - WORK AREA TO HOLD COMMAND LINE
1044      ; GCMLN - LENGTH OF LINE READ
1045      ; GCMPNT - POINTER TO COMMAND LINE (SET TO POINT TO BEGINNING)
1046      ;
1047      ; SEE 'FIND' SUBROUTINE FOR THE USE AND UPDATING OF THESE FIELDS.
1048      ;
1049      ; GCONLY:
1050      012510 012700 000020'      MOV      #GCMBUF,R0      ;POINT TO COMMAND LINE BUFFER
1051      012514 012701 000120'      MOV      #0,R1      ;NUMBER OF BYTES IN BUFFER
1052      012520 112720 000040'      1$:      MOVB     #40,(R0)+      ;CLEAR COMMAND LINE
1053      012524 005301              DEC      R1      ;FINISHED?
1054      012526 001374              BNE     1$      ;NO
1055      ;
1056      012530              GCML$      #GCMBLK
1057      012530 012700 007614'      MOV      #GCMBLK,R0
1058      012534 005060 000142'      CLR      G.PSDS(R0)
1059      012540 004767 000000G      JSR      PC,,GCML1
1060      012544 016067 000146 165370      MOV      G.CMLD(R0),GCMLN ;SAVE LENGTH
1061      012552 012767 000020' 165364      MOV      #GCMBUF,GCMPNT ;INIT COMMAND LINE POINTER
1062      012560 012701 000020'      MOV      #GCMBUF,R1      ;POINT R1 TO COMMAND LINE
1063      012564 000207              RTS      PC

```

Approved For Release 2005/07/11 : CIA-RDP85-00514R000200020001-3

012742	005046			CLR	-(SP)	
012744	005046			CLR	-(SP)	
012746	012746	000001		MOV	#LUN.TT, -(SP)	
012752	012746	000000G		MOV	#ID.DET, -(SP)	
012756	012746			MOV	(PC)+, -(SP)	
012760	003	014		.BYTE	3, 12	
012762	104377			EMT	+0<377>	
1098	012764	000207		RTS	PC	
1099			1\$:			
1100			:			
1101			:	AST		
1102			:			
1103			:			
1104	012766		AST:			
1105	012766	012667	165154	MOV	(SP)+, ASTWRD	:GET CHAR OFF STACK
1106	012772	042767	000004 165014	BIC	#LOOP, BASE	:CLEAR LOOP FLAG
1107	013000			ASTX\$S		
	013000	012746		MOV	(PC)+, -(SP)	
	013002	163	001	.BYTE	115, 1	
	013004	104377		EMT	+0<377>	

Approved For Release 2005/07/17 : CIA-RDP85-00514R000200020001-3

```

1155      ;
1156      ;
1157      ;
1158      ;
1159      ;
1160      ;
1161      ;
1162      ;
1163      ;
1164      ;
1165      ;
1166      ;
1167      ;
1168      ;
1169      ;
1170      ;
1171      ;
1172      ;
1173      ;
1174      013100      ;
1175      013100      010246      ;
1176      013102      016701      165034      ;
1177      013106      001440      ;
1178      013110      016702      165030      ;
1179      013114      122712      000040      ;
1180      013120      001403      ;
1181      013122      122712      000054      ;
1182      013126      001004      ;
1183      013130      005202      ;
1184      013132      005301      ;
1185      013134      001367      ;
1186      013136      000424      ;
1187      ;
1188      013140      010246      ;
1189      013142      005000      ;
1190      013144      122712      000040      ;
1191      013150      001407      ;
1192      013152      122712      000054      ;
1193      013156      001404      ;
1194      013160      005202      ;
1195      013162      005200      ;
1196      013164      005301      ;
1197      013166      001366      ;
1198      ;
1199      013170      010267      164750      ;
1200      013174      010167      164742      ;
1201      013200      012601      ;
1202      013202      012602      ;
1203      013204      000241      ;
1204      013206      000207      ;
1205      ;
1206      013210      012602      ;
1207      013212      000261      ;
1208      013214      000207      ;

```

FIND THE NEXT NON-BLANK IN THE COMMAND BUFFER.
 THEN FIND THE LENGTH OF THE STRING THAT STARTS WITH THAT CHARACTER.

INPUT:
 GCMLN - NUMBER OF UNPROCESSED BYTES IN COMMAND LINE.
 GCMPT - ADDR OF NEXT UNPROCESSED POSITION IN COMMAND LINE.

OUTPUT:
 R1 -> STRING, R0 - LENGTH OF STRING.
 GCMLN, GCMPT UPDATED FOR NEXT ENTRY INTO THIS ROUTINE.

THIS ROUTINE IS DESIGNED TO BE ENTERED A NUMBER OF TIMES
 IN THE PARSING OF A COMMAND LINE. THE FIELDS GCMLN AND
 GCMPT ARE REFRESHED WHEN A NEW COMMAND LINE IS READ
 (SEE THE SUBROUTINE 'GCONLY').

```

FIND:
MOV     R2, -(SP)          ;SAVE R2
MOV     GCMLN, R1          ;#. BYTES REMAINING IN COMMAND BUFFER.
BEQ     FSECK              ;THERE ARE NONE.
MOV     GCMPT, R2          ;LOAD CURRENT POINTER.
1$:     CMPB    #40, (R2)   ;LOOK FOR A BLANK.
        BEQ     10$        ;OK, BUMP TO NEXT CHAR.
        CMPB    #'', (R2)  ;COMMA IN COMMAND LINE.
        BNE     2$         ;TREAT COMMA AS BLANK.
        INC     R2          ;BUMP POINTER.
        DEC     R1          ;SUB FROM REMAINING LENGTH.
        BNE     1$
        BR      FSECK      ;NO NON-BLANK FOUND.

2$:     MOV     R2, -(SP)   ;TEMP SAVE POINTER TO BEGINNING OF STRING
        CLR     R0          ;CLEAR CHAR COUNT.
3$:     CMPB    #40, (R2)   ;LOOK FOR A BLANK.
        BEQ     4$         ;FOUND END OF STRING.
        CMPB    #'', (R2)  ;TREAT COMMAS AS BLANKS.
        BEQ     4$
        INC     R2          ;BUMP POINTER.
        INC     R0          ;BUMP CHAR COUNT.
        DEC     R1          ;SUB FROM BYTES REMAINING.
        BNE     3$

4$:     MOV     R2, GCMPT   ;SAVE POINTER FOR NEXT TIME.
        MOV     R1, GCMLN   ;SAVE BYTES REMAINING FOR NEXT TIME.
        MOV     (SP)+, R1   ;POINTER TO BEGINNING OF STRING.
        MOV     (SP)+, R2   ;RESTORE R2.
        CLC
        RTS            PC

FSECK:  MOV     (SP)+, R2   ;RESTORE R2.
        SEC
        RTS            PC

```


Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

1261      ;
1262      ;
1263      ;      WRITE A PRINT LINE TO TT0
1264      ;
1265      ;
1266 013372 CONSOL:
1267 013372      SAVE      R0,R1
013372      MOV      R0,-(SP)
013374      MOV      R1,-(SP)

1268      ;
1269 013376 012700 000120      MOV      #00,R0      ;PRINT BUFFER BYTE COUNT
1270 013402 012701 005772      MOV      #PRINT+78,R1      ;POINT PAST END OF BUFFER
1271 013406 122741 000040      1$:      CMPB      #40,-(R1)      ;LOOK FOR A NON-BLANK
1272 013412 001003      BNE      2$      ;OK, WRITE LINE
1273 013414 005300      DEC      R0      ;DEC CHAR COUNT
1274 013416 001373      BNE      1$
1275 013420 000440      BR      ABEND2      ;NO NON-BLANKS?
1276      ;
1277 013422      2$:      QIOW$S      #IO,WVB,#LUN,TT,#EFN,1,,#STAT,,<#PRINT-2,R0>,ABEND2
013422      CLR      -(SP)
013424      CLR      -(SP)
013426      CLR      -(SP)
013430      CLR      -(SP)
013432      MOV      R0,-(SP)
013434 012746 005652      MOV      #PRINT-2,-(SP)
013440 005046      CLR      -(SP)
013442 012746 000004      MOV      #STAT,-(SP)
013446 005046      CLR      -(SP)
013450 112716 000001      MOVB      #EFN,1,(SP)
013454 012746 000001      MOV      #LUN,TT,-(SP)
013460 012746 000000G      MOV      #IO,WVB,-(SP)
013464 012746      MOV      (PC)+,-(SP)
013466      .BYTE      3,12
013470 104377      EMT      +0<377>
013472 103002      BCC      +6
013474 004767 000022      JSR      PC,ABEND2

1278      ;
1279      ;
1280 013500 012701 005654      MOV      #PRINT,R1      ;POINT TO STRING
1281 013504 112721 000040      4$:      MOVB      #40,(R1)+      ;CLEAR LINE TO BLANKS
1282 013510 005300      DEC      R0      ;DEC LOOP COUNT
1283 013512 001374      BNE      4$
1284      ;
1285 013514      RESTOR      R0,R1
013514      MOV      (SP)+,R1
013516 012600      MOV      (SP)+,R0
1286 013520 000207      RTS      PC
1287      ;
1288 013522      ABEND2: ABRT$S      #MYSELF
013522 012746 000000      MOV      #MYSELF,-(SP)
013526 017646 000000      MOV      0(SP),-(SP)
013532 062766 000002 000002      ADD      #2,2(SP)
013540 017666 000002 000002      MOV      02(SP),2(SP)
013546 012746      MOV      (PC)+,-(SP)
013550 123      .BYTE      83,3
013552 104377      EMT      +0<377>

```

```
1242.      ;
1243      ;
1244      ;   CONVERT A VALUE FROM BINARY TO PRINTABLE FORM.
1245      ;   R1 = WORD TO BE CONVERTED.
1246      ;   R5 -> PRINT LINE.
1247      ;
1248      ;
1249      UNPK::
1250      ;
1251      ;   SAVE R0,R1,R2,R3,R4
1252      ;   MOV R0,-(SP)
1253      ;   MOV R1,-(SP)
1254      ;   MOV R2,-(SP)
1255      ;   MOV R3,-(SP)
1256      ;   MOV R4,-(SP)
1257      ;
1258      ;   MOV R5,R0      ;PREPARE TO CALL SYSTEM SUBRTH
1259      ;   MOV #1,R2      ;KEEP LEADING ZEROS
1260      ;   JSR PC,$CB0MG   ;CONVERT TO OCTAL ASCII
1261      ;   MOV R0,R5      ;RESTORE PRINT LINE POINTER
1262      ;   INC R5         ;AND BUMP IT
1263      ;
1264      ;
1265      UNPKX:
1266      ;   RESTOR R0,R1,R2,R3,R4
1267      ;   MOV (SP)+,R4
1268      ;   MOV (SP)+,R3
1269      ;   MOV (SP)+,R2
1270      ;   MOV (SP)+,R1
1271      ;   MOV (SP)+,R0
1272      ;   RTS PC
```

```

1290      ;
1291      ;
1292      ;      WRITE TO TT0 AND PROMPT.
1293      ;
1294      ;
1295 013554 005267 164230  ENDTST:: INC.  ERWORD.
1296 013556 005267 164224  ENFILE:: INC.  ERWORD.
1297 013554 005267 164220  ENMEM:: INC.  ERWORD.
1298 013570 005267 164214  STOP:: INC.  ERWORD.
1299 013574 005267 164210  OUT1:: INC.  ERWORD.
1300 013600 005267 164204  ERR23:: INC.  ERWORD.
1301 013604 005267 164200  ERR22:: INC.  ERWORD.
1302 013616 005267 164174  ERR21:: INC.  ERWORD.
1303 013614 005267 164170  ERR20:: INC.  ERWORD.
1304 013620 005267 164164  ERR19:: INC.  ERWORD.
1305 013624 005267 164160  ERR18:: INC.  ERWORD.
1306 013630 005267 164154  ERR17:: INC.  ERWORD.
1307 013634 005267 164150  ERR16:: INC.  ERWORD.
1308 013640 005267 164144  ERR15:: INC.  ERWORD.
1309 013644 005267 164140  ERR14:: INC.  ERWORD.
1310 013650 005267 164134  ERR13:: INC.  ERWORD.
1311 013654 005267 164130  ERR12:: INC.  ERWORD.
1312 013660 005267 164124  ERR11:: INC.  ERWORD.
1313 013664 005267 164120  ERR10:: INC.  ERWORD.
1314 013670 005267 164114  ERR9:: INC.  ERWORD.
1315 013674 005267 164110  ERR8:: INC.  ERWORD.
1316 013700 005267 164104  ERR7:: INC.  ERWORD.
1317 013704 005267 164100  ERR6:: INC.  ERWORD.
1318 013710 005267 164074  ERR5:: INC.  ERWORD.
1319 013714 005267 164070  ERR4:: INC.  ERWORD.
1320 013720 005267 164064  ERR3:: INC.  ERWORD.
1321 013724 005267 164060  ERR2:: INC.  ERWORD.
1322 013730 005267 164054  ERR1:: INC.  ERWORD.
1323      ;
1324 013734      NESTOP:
1325 013734 005267 164050  LPTST:: INC.  ERWORD.
1326 013740 005267 164044  SELTST:: INC.  ERWORD.
1327 013744 005267 164040  BASEL:: INC.  ERWORD.
1328      000003  NEST = <.-NESTOP>4
1329      ;
1330      ;
1331      ;      USE THE INDEX ERWORD TO COUNT UP FROM THE BOTTOM OF
1332      ;      THE MESSAGE TABLE. FIND THE END OF THE MESSAGE FIRST.
1333      ;      THEN THE BEGINNING. THEN GET THE LENGTH.
1334 013750 016702 164034  MOV.  ERWORD,R2      :LOAD LOOP COUNT.
1335 013754 012701 007613  MOV.  #ASCIZ,R1      :POINT TO END OF MESSAGE TABLE.
1336 013760 105741      1$:  TSTB.  -(R1)      :LOOK FOR END OF MESSAGE.
1337 013762 001376      BNE.  1$
1338 013764 005302      DEC.  R2
1339 013766 001374      BNE.  1$      :LOOP COUNT.
1340 013770 010100      MOV.  R1,R0      :BACK UP ANOTHER MESSAGE.
1341 013772 105741      2$:  TSTB.  -(R1)      :SAVE POINTER TO END OF MESSAGE.
1342 013774 001376      BNE.  2$      :BACK UP TO BEGINNING OF MESSAGE.
1343 013776 005201      INC.  R1
1344 014000 160100      SUB.  R1,R0      :BUMP TO FIRST CHAR OF MESSAGE.
1345      ;      :R0 NOW = MESSAGE LENGTH.
1346 014002

```

```

014002 005046 CLR - (SP)
014004 005046 CLR - (SP)
014006 005046 CLR - (SP)
014010 005046 CLR - (SP)
014012 010046 MOV R0, - (SP)
014014 010146 MOV R1, - (SP)
014016 005046 CLR - (SP)
014020 012746 000004 MOV #STAT, - (SP)
014024 005046 CLR - (SP)
014026 112716 000001 MOV #EFN.1, (SP)
014032 012746 000001 MOV #LUN.TT, - (SP)
014036 012746 000000 MOV #IO.WVB, - (SP)
014042 012746 MOV (PC)+, - (SP)
014044 003 014 .BYTE 3, 12
014046 104377 EMT #0<377>
014050 103002 BCC .+6
014052 004767 JSR PC, ABEND
1347
1348 014056 CLEF$. #EFN.1
014058 012746 000001 MOV #EFN.1, - (SP)
014062 012746 MOV (PC)+, - (SP)
014064 037 002 .BYTE 31, 2
014066 104377 EMT #0<377>
1349 014070 105767 163710 TSTB STAT ;GOOD-RETURN-
1350 014074 003436 BLE ABEND ;NO-
1351
1352
1353 ISSUE GCML
1354 014076 022767 000003 163704 CMP #NEST, ERWORD ;PROMPT WITH MESSAGE
1355 014104 002427 BLT TTX ;NO JUST EXIT
1356 014106 012700 000020 MOV #GCMBUF, R0 ;POINT TO COMMAND LINE BUFFER
1357 014112 012701 000120 MOV #00, R1 ;NUMBER OF BYTES IN BUFFER
1358 014116 112720 000040 MOV #40, (R0)+ ;CLEAR COMMAND LINE
1359 014122 005301 DEC R1 ;FINISHED?
1360 014124 001374 BNE 3$ ;NO
1361
1362
1363 ISSUE GET COMMAND LINE
1364 SET UP FIELDS
1365 GCMBUF - WORK AREA TO HOLD COMMAND LINE
1366 GCMLen - LENGTH OF LINE READ
1367 GCMPNT - POINTER TO COMMAND LINE (SET TO POINT TO BEGINNING)
1368 014126 GCML$ #GCMBLK
014126 012700 007614 MOV #GCMBLK, R0
014132 005060 000142 CLR G:PSDS(R0)
014136 004767 000000 JSR PC, GCML1
1369 014142 103413 BCS ABEND
1370 014144 016067 000146 163770 MOV G:CMLD(R0), GCMLen ;SAVE LENGTH
1371 014152 012701 000020 MOV #GCMBUF, R1 ;POINT TO COMMAND BUFFER
1372 014156 012767 000020 163760 MOV #GCMBUF, GCMPNT ;INITIALIZE COMMAND BUFFER POINTER
1373 014164 005067 163620 TTX CLR ERWORD ;CLEAR ERROR NUMBER INDICATOR
1374 014170 000207 RTS PC ;AND RETURN
1375
1376 014172 ABEND: ABRT$. #MYSELF
014172 012746 000000 MOV #MYSELF, - (SP)
014176 017646 000000 MOV 0(SP), - (SP)
014202 062766 000002 000002 ADD #2, 2(SP)

```

TMAIN: M1110 27-MAR-88 13:39 PAGE 27-2

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

014210	017666	000002	000002	MOV	02(SP),2(SP)
014216	012746			MOV	(PC)+,-(SP)
014220	123	003		.BYTE	83,3
014222	104377			EMT	+0<377>
1377					
1378	010516			.END	START

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

TMAIN: MACRO:M1110 27-MAR-80 13:39 PAGE:27-3
SYMBOL TABLE

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

ABEND 014172R	BYTE20=000024	BYTE72=000110	DBLD 012074RG	FSECX 013210R
ABEND2 013522R	BYTE21=000025	BYTE73=000111	DBLDCX 012504R	F.ACTL=000076
HHIGH 000206RG	BYTE22=000026	BYTE74=000112	DBLDSX 012474R	F.ALDC=000040
ALOW 000210RG	BYTE23=000027	BYTE75=000113	DBLDXX 012506R	F.BBFS=000062
ALUCKE=000000	BYTE24=000030	BYTE76=000114	DBLDL 012102R	F.BDB=000070
ALUOE=004000	BYTE25=000031	BYTE77=000115	DBR.RD=000001	F.BGBC=000057
APLACE 000016RG	BYTE26=000032	BYTE78=000116	DB\$CPP=001457	F.BKDN=000026
ASCIZ 007613R	BYTE27=000033	BYTE79=000117	DB\$SPT=000026	F.BKDS=000020
AST 012766R	BYTE28=000034	BYTE80=000118	DB\$TPC=000023	F.BKEF=000050
ASTFLG=000020 G	BYTE29=000035	BYTE81=000121	DF1 =***** GX	F.BKP1=000051
ASTWRD 000146R	BYTE30=000036	BYTE82=000122	DISPGS=100000	F.BKST=000024
A01 =010000	BYTE31=000037	BYTE83=000123	DMAAUR=000005	F.BKVD=000064
BASE 000014RG	BYTE32=000040	BYTE84=000124	DMARRD=000003	F.CHR=000075
BASEL 013744RG	BYTE33=000041	BYTE85=000125	DMARUR=000004	F.CNTG=000034
BHIGH 000212RG	BYTE34=000042	BYTE86=000126	DPLUS 000172RG	F.DFNB=000046
BINWD 000012RG	BYTE35=000043	BYTE87=000127	DSAVE 002322RG	F.DSPT=000044
BITVAL=000000	BYTE36=000044	BYTE88=000130	DSEND 005046RG	F.DVNM=000134
BIT0 =000001	BYTE37=000045	BYTE89=000131	DTBL 000322RG	F.EFBK=000010
BIT1 =000002	BYTE38=000046	BYTE90=000132	EFN.1 =000001 G	F.EFN=000050
BIT10 =002000	BYTE39=000047	BYTE91=000133	ENBR =010000	F.EOB=000032
BIT11 =004000	BYTE40=000050	BYTE92=000134	ENDMEM 013564RG	F.ERR=000052
BIT12 =010000	BYTE41=000051	BYTE93=000135	ENDTST 013554RG	F.FACC=000043
BIT13 =020000	BYTE42=000052	BYTE94=000136	ENFILE 013560RG	F.FFBY=000014
BIT14 =040000	BYTE43=000053	BYTE95=000137	EN1 =***** GX	F.FNAM=000110
BIT15 =100000	BYTE44=000054	BYTE96=000140	ERR1 013730RG	F.FNB=000102
BIT2 =000004	BYTE45=000055	BYTE97=000141	ERR10 013664RG	F.FTYT=000116
BIT3 =000010	BYTE46=000056	BYTE98=000142	ERR11 013660RG	F.FVER=000120
BIT4 =000020	BYTE47=000057	BYTE99=000143	ERR12 013654RG	F.HIBK=000004
BIT5 =000040	BYTE48=000058	BYTVAL=000144	ERR13 013650RG	F.LUN=000042
BIT6 =000100	BYTE49=000059	CBKALL=001000	ERR14 013644RG	F.MBCT=000054
BIT7 =000200	BYTE50=000062	CBKCLK=000400	ERR15 013640RG	F.MBCI=000055
BIT8 =000400	BYTE51=000063	CHIGH 000216RG	ERR16 013634RG	F.MBFG=000056
BIT9 =001000	BYTE52=000064	CHLEN=000176RG	ERR17 013630RG	F.NRBD=000024
BLOW 000214RG	BYTE53=000065	CH1 =***** GX	ERR18 013624RG	F.NREC=000030
BEND 005450RG	BYTE54=000066	CLOW 000220RG	ERR19 013620RG	F.OVBS=000030
BTOVER 005452RG	BYTE55=000067	CMILUN=000002 G	ERR2 013724RG	F.RACC=000016
BTRANS 005250RG	BYTE56=000070	CNOBRE=100000	ERR20 013614RG	F.RATT=000001
BUFCX 010770R	BYTE57=000071	CNTLX=011400R	ERR21 013610RG	F.RCNM=000034
BUFCX2 011206R	BYTE58=000072	CNUM=000015 G	ERR22 013604RG	F.RCTL=000017
BUFFSA 010626RG	BYTE59=000073	COMXX=010522RG	ERR23 013600RG	F.RSI2=000002
BUFSET 011014RG	BYTE60=000074	CONSOL 013372RG	ERR3 013720RG	F.RTYP=000000
BUSFS2 011004RG	BYTE61=000075	COUNT 000166RG	ERR4 013714RG	F.SEQN=000100
BUFX 011002R	BYTE62=000076	CPCCEN=010000	ERR5 013710RG	F.SPDV=000072
BUFX2 011220R	BYTE63=000077	CPREAD=040000	ERR6 013704RG	F.SPUN=000074
BYTE0 =000000	BYTE64=000100	CPURTE=020000	ERR7 013700RG	F.STBK=000036
BYTE1 =000001	BYTE65=000101	CSADRD=000004	ERR8 013674RG	F.UNIT=000136
BYTE10=000012	BYTE66=000102	CSEQCI=100000	ERR9 013670RG	F.URBD=000020
BYTE11=000013	BYTE67=000103	CSOE=000040	ERWORD 000010R	F.VBN=000064
BYTE12=000014	BYTE68=000104	CSWRTE=000100	EX1 010620R	F.VBSZ=000060
BYTE13=000015	BYTE69=000105	CTBL 000236RG	FD.CCL=***** GX	GCMBLK 007614R
BYTE14=000016	BYTE70=000106	DATA1 000200RG	FD.REC=***** GX	GCMBUF 000020RG
BYTE15=000017	BYTE71=000107	DATA2 000202RG	FD.TTY=***** GX	GCMLEN 000142RG
BYTE16=000020		DATA3 000204RG	FIND 013100RG	GCMPTNT 000144RG
BYTE17=000021		DA1 =***** GX	FNIN1 013022R	GCONLY 012510RG
BYTE18=000022			FNMTCH 013056R	GE.BIF=177775
BYTE19=000023			FNOUT1 013014R	GE.CLO=000004
BYTE2=000002			FNOUT2 013034R	GE.COM=000001

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

GE.CON=000020 MYSELF 000000R Q\$CP2=000260 RTNPT 000150RG T\$CD =000100
GE.EOF=177766 N=000144 Q\$CSC=010000 RU1 =***** GX T\$CLK=002000
GE.IND=000002 NEST=000003 Q\$CSEL=000360 SCAN=013006RG T\$DISK=000200
GE.IOR=177777 NESTOP 013734R Q\$CSET=000002 SELTST 013740RG T\$DRD=000004
GE.LC=000010 NEXTD 012456R Q\$CSP=020000 SEQ.CI=000010 T\$EMEM=010000
GE.MDE=177774 NOCRLF 012226R Q\$DMA=000001 SET 011022R T\$FSAA=000000
GE.OPR=177776 N.DID=000024 Q\$ENBK=040000 START 010516R T\$FSAB=000004
GE.RBG=177730 N.DVNM=000032 Q\$ENOP=020000 STAT 000004RG T\$FSAC=000014
GE.SIZ=000040 N.FID=000000 Q\$FAL=004000 STOP 013570RG T\$FSB2=000010
G.CMLD=000146 N.FNAM=000006 Q\$FC=000045 ST1 =***** GX T\$IB=000026
G.DPRM=000160 N.FTYP=000014 Q\$FO=000044 S\$CLR=000000 T\$IBAR=000024
G.ERR=000140 N.FVER=000016 Q\$FP=000046 S\$LA=000001 T\$IBR=020000
G.ISIZ=000020 N.NEXT=000022 Q\$HBF=000002 S\$OB=000005 T\$IBF=040000
G.LPDL=000060 N.STAT=000020 Q\$ICP=000006 S\$OR=000006 T\$ICD=000040
G.MODE=000141 N.UNIT=000034 Q\$IHB=000003 S\$OX=000004 T\$MODE=004000
G.PSDS=000142 OFFSET 012006RG Q\$IHRL=000002 S\$SR=000007 T\$OB=000036
G.SIZE=000224 OFF6 000174RG Q\$IHRL=000002 S\$S1=000010 T\$OBE=004000
HANG 012622RG OHIGH 000226RG Q\$LBD=001000 S\$S2=000014 T\$OBF=010000
IHIGH 000222RG OLOW 000230RG Q\$LBDP=001001 S.BFHD=000020 T\$OBRA=000034
ILOW 000224RG ONCE=000010 G Q\$LBP=000001 S.FATT=000016 T\$OBWA=000032
INCVL 000160RG OUT=002000 G Q\$LCD=000003 S.FDB=000140 T\$OUTA=100000
IO.ATA=***** G Q\$LMD=000004 S.FNAM=000006 T\$RBD=000200
IO.DET=***** G Q\$LDPP=002000 S.FNB=000036 T\$RNB=000040
IO.RVB=***** G Q\$LHP=010000 S.FNBW=000017 T\$RSET=040000
IO.WVB=***** G Q\$MNC=140000 S.FNTY=000004 T\$SC=000022
KILL 012706RG PCLCX 013276R Q\$MR=000052 S.FTYP=000002 T\$SCLK=020000
LDCTRL 011222RG PCNTLX 011706R Q\$MRP=000040 S.NFEN=000020 T\$SEG1=000000
LD1 =***** GX T\$CTR=176370 TD\$CTW=176360 T\$SEG2=000001
LI1 =***** GX TD\$CTW=176360 TD\$INL=004000 T\$SEG3=000002
LOC.EN=000100 PDCX 011600R Q\$MSP=100000 TD\$MEM=000270 T\$SO=000001
LOC.WA=000000 PDCX 011612R Q\$NCLK=176000 TD\$OAR=176344 T\$UBUS=100000
LOC.WB=100000 PDVX 011572R Q\$PP=000100 TD\$OTR=176346 T\$ICLK=000400
LOOP=000004 G PLB=000010 Q\$PPSW=000320 T\$BEN=000020
LOOPR 012566RG PLC=000020 Q\$PP2=000300 UBD.IN=000020
LOOPX 012704R PLRWR=000200 Q\$QHLT=000013 UNPK 013326RG
LPTST 013734RG PLR.EN=000200 Q\$QLA=000053 UNPKX 013356R
LUN.TT=000001 G PRCNTL 011614RG Q\$QLB=000054 UPLIM 000164RG
MAREN1=000001 PRDATA 011710RG Q\$QLR=000001 WORD0=000000
MAREN2=004000 PRDX 012004R Q\$QW=000042 WORD1=000002
MARLOD=010000 PRINT 005654RG Q\$RD=000005 WORD10=000024
MAROUT=000002 PR1 =***** GX Q\$RDND=000006 WORD11=000026
MAR.LO=002000 PSEXC 013270R Q\$REBK=001000 WORD12=000030
MAR.OU=000040 QN=001000 G Q\$RNC=006000 WORD13=000032
MBKALL=001000 QP=000400 G Q\$RSC=004000 WORD14=000034
MBKCLK=000400 Q\$CR1=176420 Q\$RSET=000010 WORD15=000036
MEND 000156RG Q\$CR2=176422 Q\$SM=100000 WORD16=000040
MMADDR=000100 Q\$CLBR=176424 Q\$SP=000120 WORD17=000042
MMLEFT=000002 Q\$ATTN=000100 Q\$SP2=000340 WORD18=000044
MMOE=000004 Q\$BCL=000001 Q\$STP=000020 WORD19=000046
MMURTE=000010 Q\$CCCP=000040 RE1 =***** GX WORD2=000004
MNOBRE=100000 Q\$CHB=000400 RGO.EN=000200 WORD20=000050
MOVTR 012252R Q\$CHRL=000200 RGO.VA=020000 WORD21=000052
MREN1=000001 Q\$CLR=000040 RIDE=000100 G WORD22=000054
MREN2=020000 Q\$CNC=030000 RP=000200 G WORD23=000056
MSTRT 000152RG Q\$CP=000060 RSPCNT 000162RG WORD24=000060
MSTR2 000154RG Q\$CPCC=000010 RS1 010606R WORD25=000062
MSYN=000040

TMAIN: MACRO-M1110 27-MAR-80 13:39 PAGE 27-5
SYMBOL TABLE

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

WORD27= .000066	WORD45= .000132	WORD63= .000176	WORD81= .000242	WORD99= .000306
WORD28= .000070	WORD46= .000134	WORD64= .000200	WORD82= .000244	WORDVAL= .000310
WORD29= .000072	WORD47= .000136	WORD65= .000202	WORD83= .000246	WORDS= .000200RG
WORD3= .000006	WORD48= .000140	WORD66= .000204	WORD84= .000250	XTREAD= .001000
WORD30= .000074	WORD49= .000142	WORD67= .000206	WORD85= .000252	XTWRITE= .000400
WORD31= .000076	WORD5= .000012	WORD68= .000210	WORD86= .000254	\$CBDMG= .000000 GX
WORD32= .000100	WORD50= .000144	WORD69= .000212	WORD87= .000256	\$COTB= .000000 GX
WORD33= .000102	WORD51= .000146	WORD7= .000016	WORD88= .000260	\$DIV= .000000 GX
WORD34= .000104	WORD52= .000150	WORD70= .000214	WORD89= .000262	\$MUL= .000000 GX
WORD35= .000106	WORD53= .000152	WORD71= .000216	WORD9= .000022	\$\$\$= .007774R
WORD36= .000110	WORD54= .000154	WORD72= .000220	WORD90= .000264	\$\$\$ARG= .000002
WORD37= .000112	WORD55= .000156	WORD73= .000222	WORD91= .000266	\$\$\$T1= .000067
WORD38= .000114	WORD56= .000160	WORD74= .000224	WORD92= .000270	\$\$\$T2= .000027
WORD39= .000116	WORD57= .000162	WORD75= .000226	WORD93= .000272	.FSRCB= .000000 G
WORD4= .000010	WORD58= .000164	WORD76= .000230	WORD94= .000274	.GCMLI= .000000 G
WORD40= .000120	WORD59= .000166	WORD77= .000232	WORD95= .000276	...PC1= .010320R
WORD41= .000122	WORD6= .000014	WORD78= .000234	WORD96= .000300	...PC2= .010474R
WORD42= .000124	WORD60= .000170	WORD79= .000236	WORD97= .000302	...PC3= .010320R
WORD43= .000126	WORD61= .000172	WORD8= .000020	WORD98= .000304	...TPC= .000020
WORD44= .000130	WORD62= .000174	WORD80= .000240		

. ABS. 000000 000
014224 001
\$\$\$FSR1 002040 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 8170 WORDS (32 PAGES)
DYNAMIC MEMORY: 9140 WORDS (35 PAGES)
ELAPSED TIME: 00:01:42
TMAIN, TMAIN/SP=C20, 1JIM, C20, 1JMAIN.

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

1      .TITLE: TREG
2      .PSECT: TREG
3      .LIST: MEB
4
5      ;
6      ;
7      ; TERM DETECTOR: 'MANUAL' DEBUGGING AIDS
8      ; REGISTER EXERCISES.
9      ;
10     ;
11     ; COMMANDS:
12     ; ST      STORE INTO A TD REGISTER
13     ; RE      READ FROM A TD REGISTER.
14     ;
15     ; ONCE A COMMAND HAS BEEN EXECUTED (OR AN ERROR ENCOUNTERED)
16     ; THIS MODULE RETURNS CONTROL TO THE MODULE TMAIN AT LOCATION
17     ; 'COMXX'.
18     ;
19     ;
20     ; STORE
21     ; TABLE OF REGISTER MNEMONICS AND ASSOCIATED ROUTINE
22     ; ADDRESSES.
23     .ST1TBL:
24     000000      101      101      .ASCII: /AA/      :FSA A MEMORY ADDRESS REGISTER
25     000002      000224      .WORD: STA
26     000004      102      101      .ASCII: /BA/      :FSA B MEMORY ADDRESS REGISTER
27     000006      000234      .WORD: STB
28     000010      103      101      .ASCII: /CA/      :FSA C MEMORY ADDRESS REGISTER
29     000012      000244      .WORD: STC
30     000014      111      102      .ASCII: /IB/      :INPUT BUFFER MEMORY ADDRESS REGISTER
31     000016      000276      .WORD: STI
32     000020      117      102      .ASCII: /OB/      :OUTPUT BUFFER MEMORY ADDRESS REGISTER
33     000022      000322      .WORD: STO
34     000024      102      124      .ASCII: /BT/      :BYTE TRANSLATOR MEMORY ADDRESS REGISTER
35     000026      000346      .WORD: STBT
36     000030      123      103      .ASCII: /SC/      :SECTOR COUNTER
37     000032      000372      .WORD: STSC
38     000034      103      122      .ASCII: /CR/      :CONTROL REGISTER
39     000036      000416      .WORD: STCR
40     000010      000010      ST1LN: == <.-ST1TBL>/4
41     ;
42     ;
43     ; READ
44     .RE1TBL:
45     000040      101      101      .ASCII: /AA/      :FSA A MEMORY ADDRESS REGISTER
46     000042      000526      .WORD: REAA
47     000044      102      101      .ASCII: /BA/      :FSA B MEMORY ADDRESS REGISTER
48     000046      000542      .WORD: REBA
49     000050      103      101      .ASCII: /CA/      :FSA C MEMORY ADDRESS REGISTER
50     000052      000556      .WORD: RECA
51     000054      111      102      .ASCII: /IB/      :INPUT BUFFER MEMORY ADDRESS REGISTER
52     000056      000572      .WORD: REIA
53     000060      117      102      .ASCII: /OB/      :OUTPUT BUFFER MEMORY ADDRESS REGISTER
54     000062      000614      .WORD: REDA
55     000064      102      124      .ASCII: /BT/      :BYTE TRANSLATOR MEMORY ADDRESS REGISTER
56     000066      000636      .WORD: RETA
57     000070      123      103      .ASCII: /SC/      :SECTOR COUNTER
58     000072      000660      .WORD: RESC

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

58 000074      123      127      .ASCII /SW/      :STATUS WORD
59 000076 000710'      .WORD RESW
60 000100      103      122      .ASCII /CR/
61 000102 000702'      .WORD RECR      :CONTROL REGISTER
62 000104      115      123      .ASCII /MS/
63 000106 000716'      .WORD REMS      :MISC SIGNALS
64 000110      121      122      .ASCII /QR/
65 000112 000740'      .WORD REQR      :TD OUTPUT BUFFER (TO QR)
66      000013      RE1LN == <.-RE1TBL>4
67
68
69
70
71
72
73
74
75 000114      ST1::
76 000114 004767 000000G      JSR PC.FIND      :LOCATE OPERAND IN COMMAND LINE
77 000120 103003      BCC 1$      :OK, CONTINUE
78 000122 004767 000000G      JSR PC.ERR3      :MISSING OPERAND
79 000126 000543      BR ST1X      :EXIT
80 000130 004767 000000G      1$: JSR PC.PACK      :CONVERT COMMAND LINE VALUE TO BINARY
81 000134 103003      BCC 2$      :CONVERSION SUCCESSFUL
82 000136 004767 000000G      JSR PC.ERR4      :INVALID NUMERIC VALUE
83 000142 000535      BR ST1X
84
85
86
87
88 000144 004767 000000G      2$: JSR PC.FIND      :LOCATE REG MNEMONIC
89 000150 103003      BCC 3$      :OK, CONTINUE
90 000152 004767 000000G      JSR PC.ERR3      :MISSING OPERAND
91 000156 000527      BR ST1X
92
93
94
95
96 000160 012700 000010      3$: MOV #ST1LN,R0      :NUMBER OF TABLE ENTRIES
97 000164 012702 000000'      MOV #ST1TBL,R2      :POINT TO TABLE
98 000170 004767 000000G      JSR PC.SCAN      :MATCH AGAINST COMMAND LINE
99 000174 103003      BCC 4$      :MATCH WAS MADE
100 000176 004767 000000G      JSR PC.ERR5      :INVALID REGISTER MNEMONIC
101 000202 000515      BR ST1X
102
103
104
105
106
107
108
109
110
111 000204 010167 000000G      4$: MOV R1,R1RNP      :SAVE POINTER
112 000210 004767 000000G      JSR PC.LOOPR      :LOOP ON TEST
113 000214 016701 000000G      STIN: MOV R1RNP,R1      :POINT TO ROUTINE
114 000220 000171 000000      JMP @R1      :JUMP TO ROUTINE

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

115      ;
116      ;
117      ;
118      ;
119 000224 012767 000000 000000G STA:: MOV. #T$FSA, APLACE ; SELECT FSA A.
120 000232 000407          ;          STOR          ; MOVE SELECTION TO TD.
121 000234 012767 000004 000000G STB:: MOV. #T$FSA, APLACE ; SELECT FSA B.
122 000242 000403          ;          STOR          ;
123 000244 012767 000014 000000G STC:: MOV. #T$FSA, APLACE ; SELECT FSA C.
124      ;
125 000252 052767 020000 000000G STOR: BIS. #T$SCLK, APLACE ; START CLOCK.
126 000260 016767 000000G 176360 MOV. APLACE, TD$CTW. ; MOVE TO CONTROL REG.
127 000266 016767 000000G 176362 MOV. BINWD, TD$TAW. ; MOVE VALUE FROM COMMAND LINE.
128 000274 000454          ;          BR          STLP ; CHECK LOOP FLAG.
129      ;
130      ;
131      ;
132 000276 012767 000040 176360 STI:: MOV. #T$ICD, TD$CTW. ; ENABLE INTERFACE AND CONTROL.
133 000304 012767 000024 176362 MOV. #T$IBAR, TD$TAW.
134 000312 016767 000000G 176364 MOV. BINWD, TD$TDW.
135 000320 000442          ;          BR          STLP ; TEST LOOP FLAG.
136      ;
137      ;
138      ;
139 000322 012767 000040 176360 STO:: MOV. #T$ICD, TD$CTW. ; ENABLE INTERFACE AND CNTL.
140 000330 012767 000032 176362 MOV. #T$OBWA, TD$TAW. ; SELECT OUTPUT BUFFER.
141 000336 016767 000000G 176364 MOV. BINWD, TD$TDW. ; MOVE WORD TO TRANSFER REG.
142 000344 000430          ;          BR          STLP
143      ;
144      ;
145      ;
146 000346 012767 000040 176360 STBT:: MOV. #T$ICD, TD$CTW. ; ENABLE INTERFACE AND CONTROL.
147 000354 012767 000030 176362 MOV. #T$BTAR, TD$TAW. ; SELECT BYTE TRANSLATOR.
148 000362 016767 000000G 176364 MOV. BINWD, TD$TDW. ; TRANSFER DATA WORD.
149 000370 000416          ;          BR          STLP
150      ;
151      ;
152      ;
153 000372 012767 000040 176360 STSC:: MOV. #T$ICD, TD$CTW. ; ENABLE INTERFACE AND CONTROL.
154 000400 012767 000022 176362 MOV. #T$SC, TD$TAW. ; SELECT SECTOR COUNTER.
155 000406 016767 000000G 176364 MOV. BINWD, TD$TDW. ; TRANSFER DATA WORD.
156 000414 000404          ;          BR          STLP
157      ;
158      ;
159      ;
160 000416 016767 000000G 176360 STCR:: MOV. BINWD, TD$CTW. ; MOVE TO CONTROL REG.
161 000424 000400          ;          BR          STLP
162      ;
163      ;
164 000426 032767 000000G 000000G STLP: BIT. #LOOP, BASE. ; LOOP FLAG ON.
165 000434 001267          ;          BNE. STIN ; YES, REPEAT.
166      ;
167 000436          ;
168 000436 004767 000000G STIX: JSR. PC, KILL. ; KILL AST. (IF THERE WAS ONE)
169 000442 000167 000000G JMP. COMMX.

```

```
171 ;
172 ;
173 ; READ.
174 ; PERFORM SECOND-LEVEL PARSING.
175 ; EG. IN THE COMMAND:
176 ; >RE CR.
177 ; PARSE THE 'CR'
178 ;
179 ;
180 000446 RE1::
181 000446 004767 000000G JSR PC,FIND ;LOCATE THE REGISTER MNEMONIC
182 000452 103003 BCC 1$ ;FOUND ONE
183 000454 004767 000000G JSR PC,ERR3 ;MISSING OPERAND
184 000460 000574 BR RE1X
185 ;
186 ;
187 ; MATCH THE REGISTER MNEMONIC FROM THE COMMAND LINE
188 ; AGAINST A TABLE OF VALID MNEMONICS.
189 000462 012700 000013 1$: MOV #RE1LN,R0 ;NUMBER OF TABLE ENTRIES
190 000466 012702 000040 MOV #RE1TBL,R2 ;POINT TO TABLE
191 000472 004767 000000G JSR PC,SCAN ;MATCH AGAINST COMMAND LINE
192 000476 103003 BCC 2$ ;MATCH WAS MADE
193 000500 004767 000000G JSR PC,ERR5 ;INVALID REGISTER MNEMONIC
194 000504 000562 BR RE1X
195 ;
196 ;
197 ; SAVE THE POINTER TO THE ROUTINE ASSOCIATED WITH THE
198 ; REGISTER. R1 -> ROUTINE ADDRESS.
199 ; CALL ROUTINE TO SCAN THE COMMAND LINE FOR A LOOP
200 ; INDICATOR. EG:
201 ; >RE CR L
202 ; LOOP FLAG WILL BE SET IF INDICATOR IS PRESENT.
203 ; JUMP TO ROUTINE TO READ REGISTER.
204 000506 010167 000000G 2$: MOV R1,R1NPT ;SAVE ROUTINE POINTER
205 000512 004767 000000G JSR PC,LOOPR ;LOOP ON TEST?
206 000516 016701 000000G REIN: MOV RTNPT,R1 ;POINT TO ROUTINE
207 000522 000171 000000 JMP @R1 ;GO THERE
208 ;
209 ;
210 ; READ FSA A MEMORY ADDRESS REGISTER.
211 ;
212 000526 REAA::
213 000526 012767 000000 176360 MOV #T$FSA,TD$CTW ;SELECT FSA A
214 000534 016701 176372 MOV TD$TAR,R1 ;GET REG CONTENTS
215 000540 000523 BR REPUT ;PRINT
216 ;
217 ;
218 ; READ FSA B MEMORY ADDRESS REGISTER.
219 000542 REBA::
220 000542 012767 000004 176360 MOV #T$FSB,TD$CTW ;SELECT FSA B
221 000550 016701 176372 MOV TD$TAR,R1 ;GET REG CONTENTS
222 000554 000515 BR REPUT ;AND PRINT
223 ;
224 ;
225 ; READ FSA C MEMORY ADDRESS REGISTER.
226 000556 RECA::
227 000556 012767 000014 176360 MOV #T$FSC,TD$CTW ;SELECT FSA C
```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
228 000564 016701 176372 MOV TD$TAR,R1 ;GET REG CONTENTS
229 000570 000507 BR REPUT ;PRINT
230
231
232
233 000572 REIA::
234 000572 012767 000040 176360 MOV #T$ICD,TD$CTW ;SELECT INTERFACE AND CONTROL
235 000600 012767 000024 176362 MOV #T$IBAR,TD$TAW ;SELECT INPUT BUF REG
236 000606 016701 176374 MOV TD$TDR,R1 ;GET REG CONTENTS
237 000612 000476 BR REPUT ;PRINT
238
239
240 READ OUTPUT BUFFER MEMORY ADDRESS REGISTER
241 000614 REOA::
242 000614 012767 000040 176360 MOV #T$ICD,TD$CTW ;SELECT INTERFACE AND CONTROL
243 000622 012767 000034 176362 MOV #T$OBRA,TD$TAW ;SELECT OUTPUT BUFFER MAR
244 000630 016701 176374 MOV TD$TDR,R1 ;GET REG CONTENTS
245 000634 000465 BR REPUT ;PRINT
246
247
248 READ BYTE TRANSLATOR MEMORY ADDRESS REGISTER
249 000636 RETA::
250 000636 012767 000040 176360 MOV #T$ICD,TD$CTW ;SELECT INTERFACE AND CONTROL
251 000644 012767 000030 176362 MOV #T$BTAR,TD$TAW ;SELECT BYTE TRANSLATOR
252 000652 016701 176374 MOV TD$TDR,R1 ;GET BYTE TRANS MAR
253 000656 000454 BR REPUT ;PRINT
254
255
256 READ SECTOR COUNTER
257 000660 RESC::
258 000660 012767 000040 176360 MOV #T$ICD,TD$CTW ;SELECT INTERFACE AND CONTROL
259 000666 012767 000022 176362 MOV #T$SC,TD$TAW ;SELECT SECTOR COUNTER
260 000674 016701 176374 MOV TD$TDR,R1 ;READ SECTOR COUNTER
261 000700 000443 BR REPUT ;AND PRINT
262
263
264 READ CONTROL REGISTER
265 000702 RECR::
266 000702 016701 176370 MOV TD$CTR,R1 ;LOAD CONTROL REG
267 000706 000440 BR REPUT ;PRINT IT
268
269
270 READ STATUS REGISTER
271 000710 RESW::
272 000714 000435 MOV TD$SW,R1 ;LOAD STATUS REG
273 BR REPUT ;PRINT IT
274
275
276 READ MISC CONTROL SIGNALS
277 000716 REMS::
278 000716 012767 000040 176360 MOV #T$ICD,TD$CTW ;SELECT INTERFACE AND CONTROL
279 000724 012767 000030 176362 MOV #T$BTAR,TD$TAW ;LOAD MISC REG
280 000732 016701 176374 MOV TD$TDR,R1 ;READ MISC REG
281 000736 000424 BR REPUT
282
283 PRINT TD OUTPUT TO QR
284 000740 REQR::
```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

285 000740 016701 176344      MOV.    TD$OAR,R1      ;READ OUTPUT AVAILABLE REG.
286 000744 032701 100000      BIT.     *T$OUTA,R1    ;IS OUTPUT AVAILABLE
287 000750 001006              BNE.     1$              ;YES, PRINT
288 000752 012767 046505 000000G. MOV.     *EM,PRINT.    ;EMPTY
289 000760 004767 000000G.      JSR.     PC,CONSOL.    ;WRITE MESSAGE
290 000764 000426              BR.       RELP              ;TEST LOOP
291 000766 012705 000000G.      1$:    MOV.     *PRINT,R5      ;POINT TO PRINT LINE
292 000772 016701 176346      MOV.     TD$OTR,R1    ;READ OUTPUT
293 000776 004767 000000G.      JSR.     PC,UNPK.      ;CONVERT
294 001002 004767 000000G.      JSR.     PC,CONSOL.    ;PRINT IT
295 001006 000415              BR.       RELP              ;TEST LOOP
296
297
298
299
300 001010      REPUT:
301 001016 001011      BIT.     #ONCE,BASE.    ;CONTENTS ALREADY PRINTED ONCE
302 001020 052767 000000G 000000G. BNE.     RELP              ;YES, SKIP SUBSEQUENT PRINTINGS
303 001026 012705 000000G.      BIS.     #ONCE,BASE.    ;SET FLAG FOR LINE PRINTED
304 001032 004767 000000G.      MOV.     *PRINT,R5      ;SET UP POINTER TO PRINT LINE
305 001036 004767 000000G.      JSR.     PC,UNPK.      ;CONVERT
306
307 001042 032767 000000G 000000G. RELP:  JSR.     PC,CONSOL.    ;WRITE TO TTY
308 001050 001222      ;
309
310 001052      REIX:
311 001052 042767 000000G 000000G. BIC.     #ONCE,BASE.    ;CLEAR LOCAL FLAG
312 001060 004767 000000G.      JSR.     PC,KILL.      ;KILL LAST (IF THERE WAS ONE)
313 001064 000167 000000G.      JMP.     COMXX
314
315      .END

```

ALUCKE = 040000	BYTE4 = 000004	BYTE91 = 000133	N = 000144	Q\$QL = 000043
ALUDE = 004000	BYTE40 = 000050	BYTE92 = 000134	ONCE = ***** GX	Q\$QLA = 000053
APLACE = ***** GX	BYTE41 = 000051	BYTE93 = 000135	PACK = ***** GX	Q\$QLB = 000054
A01 = 010000	BYTE42 = 000052	BYTE94 = 000136	PLB = 000010	Q\$QLR = 000001
BASE = ***** GX	BYTE43 = 000053	BYTE95 = 000137	PLC = 000020	Q\$QJW = 000042
BINWD = ***** GX	BYTE44 = 000054	BYTE96 = 000140	PLD = 000030	Q\$RDCD = 000005
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	PLRWR = 000200	Q\$RDMD = 000006
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	PLR.EN = 000200	Q\$REBK = 001000
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	PRINT = ***** GX	Q\$RNC = 006000
BIT10 = 000200	BYTE48 = 000060	BYTVAL = 000144	Q\$CR1 = 176420	Q\$RSC = 004000
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	Q\$CR2 = 176422	Q\$RSET = 000010
BIT12 = 010000	BYTE5 = 000005	CBKCLK = 000400	Q\$LBR = 176424	Q\$SM = 100000
BIT13 = 020000	BYTE50 = 000062	CNOBRE = 100000	Q\$ATTN = 000100	Q\$SP = 000120
BIT14 = 040000	BYTE51 = 000063	COMXX = ***** GX	Q\$BCL = 000001	Q\$SP2 = 000340
BIT15 = 100000	BYTE52 = 000064	CONSOL = ***** GX	Q\$CCCP = 000040	REAA = 000526RG 002
BIT2 = 000004	BYTE53 = 000065	CPCCEN = 010000	Q\$CHB = 000400	REBA = 000542RG 002
BIT3 = 000010	BYTE54 = 000066	CPREAD = 040000	Q\$CHRL = 000200	RECA = 000556RG 002
BIT4 = 000020	BYTE55 = 000067	CPWRTE = 020000	Q\$CLR = 000040	RECR = 000702RG 002
BIT5 = 000040	BYTE56 = 000070	CSADRD = 000004	Q\$CNC = 030000	REIA = 000572RG 002
BIT6 = 000100	BYTE57 = 000071	CSEDCI = 100000	Q\$CP = 000060	REIN = 000516RG 002
BIT7 = 000200	BYTE58 = 000072	CSOE = 000040	Q\$CPCC = 000010	RELP = 001042R 002
BIT8 = 000400	BYTE59 = 000073	CSWRTE = 000100	Q\$CP2 = 000260	REMS = 000716RG 002
BIT9 = 001000	BYTE6 = 000006	DBR.RD = 000001	Q\$CSC = 010000	REOA = 000614RG 002
BYTE0 = 000000	BYTE60 = 000074	DBCPP = 001457	Q\$CSEL = 000360	REPUT = 001010R 002
BYTE1 = 000001	BYTE61 = 000075	DBSPT = 000026	Q\$CSET = 000002	REOR = 000740RG 002
BYTE10 = 000012	BYTE62 = 000076	DBTPC = 000023	Q\$CSP = 020000	RESC = 000660RG 002
BYTE11 = 000013	BYTE63 = 000077	DISPGS = 100000	Q\$DNA = 000001	RESW = 000710RG 002
BYTE12 = 000014	BYTE64 = 000100	DMAAUR = 000005	Q\$ENBK = 040000	RETA = 000636RG 002
BYTE13 = 000015	BYTE65 = 000101	DMARRD = 000003	Q\$ENOP = 020000	RE1 = 000446RG 002
BYTE14 = 000016	BYTE66 = 000102	DMARUR = 000004	Q\$FAL = 004000	REILN = 000013 G
BYTE15 = 000017	BYTE67 = 000103	ENBR = 010000	Q\$FC = 000045	REITBL = 000040RG 002
BYTE16 = 000020	BYTE68 = 000104	ERR3 = ***** GX	Q\$FO = 000044	REIX = 001052R 002
BYTE17 = 000021	BYTE69 = 000105	ERR4 = ***** GX	Q\$FP = 000046	RGQ.EN = 000200
BYTE18 = 000022	BYTE7 = 000007	ERR5 = ***** GX	Q\$HBF = 000002	RGQ.VA = 020000
BYTE19 = 000023	BYTE70 = 000106	FIND = ***** GX	Q\$ICP = 000006	RTNPT = ***** GX
BYTE2 = 000002	BYTE71 = 000107	KILL = ***** GX	Q\$IHB = 000003	SCAN = ***** GX
BYTE20 = 000024	BYTE72 = 000110	LOC.EN = 000100	Q\$IHRL = 000002	SEQ.CI = 000010
BYTE21 = 000025	BYTE73 = 000111	LOC.WA = 040000	Q\$INRP = 000007	STA = 000224RG 002
BYTE22 = 000026	BYTE74 = 000112	LOC.WB = 100000	Q\$LBD = 001000	STB = 000234RG 002
BYTE23 = 000027	BYTE75 = 000113	LOOP = ***** GX	Q\$LBDP = 001001	STBT = 000346RG 002
BYTE24 = 000030	BYTE76 = 000114	LOOPR = ***** GX	Q\$LBP = 000001	STC = 000244RG 002
BYTE25 = 000031	BYTE77 = 000115	MAREN1 = 000001	Q\$LCD = 000003	STCR = 000416RG 002
BYTE26 = 000032	BYTE78 = 000116	MAREN2 = 004000	Q\$LDMD = 000004	STI = 000276RG 002
BYTE27 = 000033	BYTE79 = 000117	MARLOD = 010000	Q\$LDPP = 002000	STIN = 000214R 002
BYTE28 = 000034	BYTE8 = 000010	MAROUT = 000002	Q\$LHP = 010000	STLP = 000426R 002
BYTE29 = 000035	BYTE80 = 000120	MAR.LO = 002000	Q\$MNC = 140000	STO = 000322RG 002
BYTE3 = 000003	BYTE81 = 000121	MAR.OU = 000040	Q\$MR = 000052	STOR = 000252R 002
BYTE30 = 000036	BYTE82 = 000122	MBKALL = 001000	Q\$MRP = 000040	STSC = 000372RG 002
BYTE31 = 000037	BYTE83 = 000123	MBKCLK = 000400	Q\$MRP2 = 000240	ST1 = 000114R 002
BYTE32 = 000040	BYTE84 = 000124	MMADRD = 000100	Q\$MSC = 040000	STILN = 000010 G
BYTE33 = 000041	BYTE85 = 000125	MMLEFT = 000002	Q\$MSET = 000004	STITBL = 000000RG 002
BYTE34 = 000042	BYTE86 = 000126	MMOE = 000004	Q\$MSP = 100000	STIX = 000436R 002
BYTE35 = 000043	BYTE87 = 000127	MMWRTE = 000010	Q\$NCLK = 176000	S\$CLR = 000000
BYTE36 = 000044	BYTE88 = 000130	MNOBRE = 100000	Q\$PP = 000100	S\$Q = 000001
BYTE37 = 000045	BYTE89 = 000131	MREN1 = 000001	Q\$PPSW = 000320	S\$OB = 000000
BYTE38 = 000046	BYTE9 = 000011	MREN2 = 020000	Q\$PP2 = 000300	S\$OR = 000006
BYTE39 = 000047	BYTE90 = 000132	MSYN = 000040	Q\$QHLT = 000013	S\$QX = 000004

TREG- M1110 27-MAR-80 15:39 PAGE 6-4
SYMBOL TABLE

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

S\$SR= .000007	T\$IBAR=.000024	WORD15=.000036	WORD44=.000130	WORD73=.000222
S\$S1=.000010	T\$IBE=.020000	WORD16=.000040	WORD45=.000132	WORD74=.000224
S\$S2=.000014	T\$IBF=.040000	WORD17=.000042	WORD46=.000134	WORD75=.000226
TD\$CTR=.176370	T\$ICD=.000040	WORD18=.000044	WORD47=.000136	WORD76=.000230
TD\$CTW=.176360	T\$MODE=.004000	WORD19=.000046	WORD48=.000140	WORD77=.000232
TD\$INL=.004000	T\$OB=.000036	WORD20=.000050	WORD49=.000142	WORD78=.000234
TD\$MEM=.000270	T\$OBE=.004000	WORD21=.000052	WORD50=.000144	WORD79=.000236
TD\$QAR=.176344	T\$OBF=.010000	WORD22=.000054	WORD51=.000146	WORD80=.000240
TD\$QTR=.176346	T\$OBRA=.000034	WORD23=.000056	WORD52=.000150	WORD81=.000242
TD\$QRD=.000274	T\$OBWA=.000032	WORD24=.000060	WORD53=.000152	WORD82=.000244
TD\$SW=.176376	T\$OUTA=.100000	WORD25=.000062	WORD54=.000154	WORD83=.000246
TD\$TAR=.176372	T\$RBD0=.000200	WORD26=.000064	WORD55=.000156	WORD84=.000250
TD\$TAW=.176362	T\$RNB=.000040	WORD27=.000066	WORD56=.000160	WORD85=.000252
TD\$TDR=.176374	T\$RSET=.040000	WORD28=.000070	WORD57=.000162	WORD86=.000254
TD\$TDW=.176364	T\$SC=.000022	WORD29=.000072	WORD58=.000164	WORD87=.000256
T\$AD=.000020	T\$SCLK=.020000	WORD30=.000074	WORD59=.000166	WORD88=.000260
T\$BA=.000002	T\$SEG1=.000000	WORD31=.000076	WORD60=.000170	WORD89=.000262
T\$BD=.000010	T\$SEG2=.000001	WORD32=.000100	WORD61=.000172	WORD90=.000264
T\$BS0=.100000	T\$SEG3=.000002	WORD33=.000102	WORD62=.000174	WORD91=.000266
T\$BT=.000020	T\$SO=.000001	WORD34=.000104	WORD63=.000176	WORD92=.000270
T\$BTAR=.000030	T\$UBUS=.100000	WORD35=.000106	WORD64=.000200	WORD93=.000272
T\$BTD=.002000	T\$1CLK=.000400	WORD36=.000110	WORD65=.000202	WORD94=.000274
T\$CD=.000100	T\$OBEN=.000020	WORD37=.000112	WORD66=.000204	WORD95=.000276
T\$CLK=.002000	UBD.IN=.000020	WORD38=.000114	WORD67=.000206	WORD96=.000300
T\$DISK=.000200	UNPK=.***** GX	WORD39=.000116	WORD68=.000210	WORD97=.000302
T\$DRD=.000004	WORD0=.000000	WORD40=.000120	WORD69=.000212	WORD98=.000304
T\$EMEM=.010000	WORD1=.000002	WORD41=.000122	WORD70=.000214	WORD99=.000306
T\$FSAA=.000000	WORD10=.000024	WORD42=.000124	WORD71=.000216	WRDVAL=.000310
T\$FSAB=.000004	WORD11=.000026	WORD43=.000126	WORD72=.000220	XTREAD=.001000
T\$FSAC=.000014	WORD12=.000030			XTWRITE=.000400
T\$FSB2=.000010	WORD13=.000032			
T\$IB=.000026	WORD14=.000034			

. ABS. 000000 000
000000 001
TREG. 001070 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3266 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:45
TREG, TREG/SP=[20,1]IM,[20,1]TREG

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

58 000060 004767 000000G..... JSR. PC.FIND. ;LOCATE MEMORY MNEMONIC IN COMMAND LINE.
59 000064 103004 BCC. 1$ ;OK, CONTINUE.
60 000066 004767 000000G. JSR. PC.ERR3 ;"MISSING OPERAND"
61 000072 000167 001246 JMP. LDIX ;EXIT.
62. ;
63 000076 012700 000006 1$: MOV. #LDILN,R0 ;NUMBER OF TABLE ENTRIES.
64 000102 012702 000000* MOV. #LDITBL,R2. ;POINT TO TABLE.
65 000106 004767 000000G. JSR. PC.SCAN. ;MATCH AGAINST COMMAND LINE.
66 000112 103004 BCC. 2$ ;MATCH WAS MADE.
67 000114 004767 000000G. JSR. PC.ERR6 ;"INVALID MEMORY MNEMONIC"
68 000120 000167 001220 JMP. LDIX
69 ;
70 ;
71 ;
72 ;
73 ;
74 ;
75 ;
76 ;
77 ;
78 ;
79 ;
80 ;
81 ;
82 ;
83 ;
84 ;
85 ;
86 ;
87 ;
88 ;
89 ;
90 ;
91 000124 010167 000000G. 2$: MOV. R1,RTHPT. ;SAVE POINTER.
92 000130 004767 000000G. JSR. PC.FIND. ;LOCATE START ADDR IN COMMAND LINE.
93 000134 103004 BCC. 3$ ;OK, CONTINUE.
94 000136 004767 000000G. JSR. PC.ERR3 ;"MISSING OPERAND"
95 000142 000167 001176 JMP. LDIX ;EXIT.
96 000146 004767 000000G. 3$: JSR. PC.PACK. ;CONVERT COMMAND LINE VALUE TO BINARY.
97 000152 103004 BCC. 4$ ;CONVERSION SUCCESSFUL.
98 000154 004767 000000G. JSR. PC.ERR4 ;"INVALID NUMERIC VALUE"
99 000160 000167 001160 JMP. LDIX
100 ;
101 000164 016767 000000G.000000G. 4$: MOV. BINWD,MSTR1. ;SAVE LOADING START ADDRESS.
102 000172 016767 000000G.000000G. MOV. BINWD,MSTR2. ;SAVE IT TWICE (FOR REFRESH ON LOOP)
103 000200 012767 177777 000000G. MOV. #-1,MEND. ;INIT END ADDRESS.
104 ;
105 ;
106 ;
107 ;
108 ;
109 ;
110 000206 004767 000000G. JSR. PC.FIND. ;SCAN COMMAND LINE
111 000212 103004 BCC. 5$ ;SOMETHING THERE
112 000214 052767 000000G.000000G. BIS. #RP,BASE. ;SIGNAL TO REPEAT PROMPT
113 000222 000437 BR 9$ ;JUMP TO RTN.
114 ;

```

```

115 000224 122711 000114      5$:  GMPB.  #*L.(R1)      :LOOP INDICATOR.
116 000230 001006              BNE.    6$          :NO, MUST BE UPPER ADDRESS.
117 000232 016767 000000G.000000G.  MOV.   MSTRT,MEND.  :LOWER ADDRESS BECOMES UPPER ADDRESS.
118 000240 004767 000000G.      JSR.    PC,HANG.    :HOW TO STOP LOOP.
119 000244 000426              BR.     9$          :JUMP TO RTN.
120
121 000246 004767 000000G.      6$:  JSR.    PC,PACK.    :CONVERT UPPER ADDRESS.
122 000252 103004              BCC.    7$          :OK, CONTINUE.
123 000254 004767 000000G.      JSR.    PC,ERR4    :INVALID NUMERIC VALUE?
124 000260 000167 001060      JMP.    LD1X      :AND EXIT.
125
126
127
128
129 000264 016767 000000G.000000G.  7$:  MOV.   BINWD,MEND.  :SET UP ENDING ADDRESS.
130 000272 004767 000000G.      JSR.    PC,FIND    :CHECK FOR LOOP INDICATOR.
131 000276 103411              BCS.    9$          :NO LOOP
132 000300 122711 000114      CMPB.   #*L.(R1)    :CORRECT INDICATOR
133 000304 001404              BEQ.    0$          :YES, CONTINUE.
134 000306 004767 000000G.      JSR.    PC,ERR1    :NO, PUT OUT MESSAGE.
135 000312 000167 001026      JMP.    LD1X      :AND EXIT.
136 000316 004767 000000G.      8$:  JSR.    PC,HANG.    :HOW TO STOP LOOP.
137
138 000322 016701 000000G.      9$:  MOV.   RTNPT,R1    :POINT TO ROUTINE.
139 000326 000171 000000G.      JMP.    @R1      :JUMP TO ROUTINE.
140
141
142
143
144 000332
145 000332 012767 000000G.000000G.  LDAM::
146 000340 016746 000000G.      MOV.   #T$FSA,A,PLAC  :SELECT FSA A FOR SUBRTN.
147 000344 016746 000000G.      MOV.   AHIGH,-(SP)   :UPPER MEMORY LIMIT.
148 000350 004767 000000G.      MOV.   ALOW,-(SP)   :LOWER LIMIT.
149 000354 103402      JSR.    PC,BUFFSA.  :CHECK ADDRESS LIMITS, ETC.
150 000356 004767 000000G.      BCS.    1$          :FAILED CHECK.
151 000362 000167 000756      JSR.    PC,LDCNTL.  :LOAD FSA.
152
153
154
155 000366
156 000366 012767 000004G.000000G.  1$:  JMP.    LD1X      :AND EXIT.
157
158
159
160
161
162
163
164
165
166 000422
167 000422 012767 000014G.000000G.  LDAM::
168 000430 016746 000000G.      MOV.   #T$FSAB,APLAC  :SELECT FSA B FOR SUBRTN.
169 000434 016746 000000G.      MOV.   BHIGH,-(SP)   :UPPER MEMORY LIMIT.
170 000440 004767 000000G.      MOV.   BLOW,-(SP)   :LOWER LIMIT.
171 000444 103402      JSR.    PC,BUFFSA.  :CHECK ADDRESS LIMITS, ETC.
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

```

172 000446 004767 000000G JSR PC, LDCNTL ; LOAD FSA
173 000452 000167 000665 1$: JMP LDIX ; AND EXIT
174
175
176
177
178
179 000456 LDIB::
180 000456 012767 000200 000000G MOV #T$DISK, APLAC ; SIMULATE DISK
181 000464 016746 000000G MOV IHIGH, -(SP) ; SUPPLY UPPER MEMORY LIMIT
182 000470 016746 000000G MOV ILOW, -(SP) ; LOWER LIMIT
183 000474 004767 000000G JSR PC, BUFS2 ; PREPARE TO LOAD
184 000500 103002 100$ BCC 100$ ; NO ERRORS, CONTINUE
185 000502 000167 000636 JMP LDIX
186
187 000506 032767 000000G 100$: BIT #RP, BASE ; REPEAT PROMPT?
188 000514 001420 BEQ 2$ ; NO, ONCE ONLY
189 000516 004767 000000G 1$: JSR PC, PDATA ; READ DATA FROM COMMAND LINE
190 000522 103002 BCC 10$ ; NOT END OF MEMORY
191 000524 000167 000614 JMP LDIX ; IF END OR ERROR, EXIT
192 000530 102002 10$: BVC 20$ ; <CR> RESPONSE TO PROMPT
193 000532 000167 000606 JMP LDIX ; YES, EXIT
194 000536 016702 000000G 20$: MOV MSTR2, R2 ; LOAD IB ADDRESS
195 000542 016712 000000G MOV DATA1, (R2) ; LOAD
196 000546 066767 000000G 000000G ADD INCVAL, MSTR2 ; ADVANCE ADDRESS
197 000554 000760 BR 1$ ; REPEAT
198
199
200
201 000556 004767 000000G 2$: JSR PC, PDATA ; PROMPT ONCE
202 000562 103002 BCC 30$ ; NOT END OF MEMORY OR ERROR
203 000564 000167 000554 JMP LDIX ; IF YES, EXIT
204 000570 102004 30$: BVC 40$ ; <CR> RESPONSE TO PROMPT
205 000572 004767 000000G JSR PC, ERR3 ; YES, MISSING OPERAND
206 000576 000167 000542 JMP LDIX
207 000602 016702 000000G 40$: MOV MSTR2, R2 ; LOAD IB ADDRESS
208 000606 016712 000000G MOV DATA1, (R2)
209 000612 066767 000000G 000000G ADD INCVAL, MSTR2 ; ADVANCE ADDRESS
210 000620 026767 000000G 000000G CMP MSTR2, MEND ; HAS UPPER MEMORY LIMIT BEEN REACHED
211 000626 101765 BLOS 40$ ; NO, CONTINUE
212 000630 032767 000000G 000000G BIT #LOOP, BASE ; REPEAT COMMAND
213 000636 001002 BNE 50$ ; YES, CONTINUE
214 000640 000167 000500 JMP LDIX ; ELSE EXIT
215 000644 016767 000000G 50$: MOV MSTR2, MSTR2 ; REINITIALIZE ADDRESS
216 000652 000753 BR 40$
217
218
219
220
221 000654 LDDB::
222 000654 012767 000040 000000G MOV #T$ICD, APLAC ; SET INTERFACE AND CONTROL FOR SUBRTN
223 000662 016746 000000G MOV OHIGH, -(SP) ; SUPPLY UPPER MEMORY LIMIT
224 000666 016746 000000G MOV OLOW, -(SP) ; LOWER LIMIT
225 000672 004767 000000G JSR PC, BUFS2 ; PREPARE FOR LOAD
226 000676 103002 BCC 100$ ; OK, CONTINUE
227 000700 000167 000440 JMP LDIX ; ERROR, EXIT
228

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

229 000704 332767 000000G-000000G-100$: BIT #RP,BASE :REPEAT PROMPT.?
230 000712 001430 BEQ 2$ :NO, ONCE ONLY
231 000714 004767 1$: JSR PC,PDATA :READ DATA FROM COMMAND LINE
232 000720 103002 BCC 10$ :NOT END OF MEMORY
233 000722 000167 000416 JMP LD1X :IF END OR ERROR, EXIT
234 000726 102002 10$: BVC 20$ :<CR> RESPONSE TO PROMPT
235 000730 000167 000410 JMP LD1X :YES, EXIT
236 000734 012767 000032 176362 20$: MOV #T$OBWA,TD$TAW
237 000742 016767 000000G-176364 MOV MSTR2,TD$TDW
238 000750 012767 000036 176362 MOV #T$OB,TD$TAW
239 000756 016767 000000G-176364 MOV DATA1,TD$TDW
240 000764 066767 000000G-000000G ADD INCVAL,MSTR2 :ADVANCE ADDRESS
241 000772 000750 BR 1$ :REPEAT
242 :
243 :
244 :
245 000774 004767 000000G 2$: JSR PC,PDATA :PROMPT ONCE
246 001000 103002 BCC 30$ :NOT END OF MEMORY OR ERROR
247 001002 000167 000336 JMP LD1X :IF YES, EXIT
248 001006 102004 30$: BVC 40$ :<CR> RESPONSE TO PROMPT.?
249 001010 004767 000000G JSR PC,ERR3 :YES, MISSING OPERAND
250 001014 000167 000324 JMP LD1X
251 001020 012767 000032 176362 40$: MOV #T$OBWA,TD$TAW
252 001026 016767 000000G-176364 MOV MSTR2,TD$TDW
253 001034 012767 000036 176362 MOV #T$OB,TD$TAW
254 001042 016767 000000G-176364 MOV DATA1,TD$TDW
255 001050 066767 000000G-000000G ADD INCVAL,MSTR2 :ADVANCE ADDRESS
256 001056 026767 000000G-000000G CMP MSTR2,MEND :HAS UPPER MEMORY LIMIT BEEN REACHED
257 001064 101755 BLOS 40$ :NO, CONTINUE
258 001066 032767 000000G-000000G BIT #LOOP,BASE :REPEAT COMMAND
259 001074 001002 BNE 50$ :YES, CONTINUE
260 001076 000167 000242 JMP LD1X :ELSE EXIT
261 001102 016767 000000G-000000G 50$: MOV MSTR1,MSTR2 :REINITIALIZE ADDRESS
262 001110 000743 BR 40$
263 :
264 :
265 :
266 :
267 001112 LDBT: :
268 001112 012767 000040 000000G MOV #T$ICD,APLACE :SET INTERFACE AND CONTROL FOR SUBRTN
269 001120 016746 000000G MOV THIGH,-(SP) :SUPPLY MEMORY UPPER LIMIT
270 001124 016746 000000G MOV TLOW,-(SP) :LOWER LIMIT
271 001130 004767 000000G JSR PC,BUFSET :PREPARE FOR LOAD
272 001134 103503 BCS LD1X :ERROR
273 :
274 001136 032767 000000G-000000G BIT #RP,BASE :REPEAT PROMPT.?
275 001144 001430 BEQ 2$ :NO, PROMPT ONLY ONCE
276 001146 004767 1$: JSR PC,PDATA :READ DATA FROM COMMAND LINE
277 001152 103002 BCC 10$ :VALID DATA PRESENT
278 001154 000167 000164 JMP LD1X :IF END OF MEMORY OR ERROR, EXIT
279 001160 102002 10$: BVC 20$ :<CR> RESPONSE TO PROMPT
280 001162 000167 000156 JMP LD1X :YES, EXIT
281 001166 012767 000030 176362 20$: MOV #T$BTAR,TD$TAW
282 001174 016767 000000G-176364 MOV MSTR2,TD$TDW :SELECT BYTE TRANS REG
283 001202 012767 000020 176362 MOV #T$BT,TD$TAW :MOVE ADDR TO TRANSFER REG
284 001210 016767 000000G-176364 MOV DATA1,TD$TDW :SELECT BYTE TRANS MEMORY
285 001216 066767 000000G-000000G ADD INCVAL,MSTR2 :LOAD BYTE TRANS
:ADVANCE ADDRESS

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

286 001224 000750          BR      1$          ;REPEAT
287                          ;
288                          ; PROMPT-ONCE-THEN-FILL-MEMORY
289                          ;
290 001226 004767 000000G    2$: JSR      PC,PDATA          ;PROMPT-ONCE
291 001232 103002          BCC      30$          ;NOT-END-OF-MEMORY-OR-ERROR
292 001234 000167 000104    JMP      LD1X          ;IF-YES-EXIT
293 001240 102004          30$: BVC      40$          ;<CR> RESPONSE-TO-PROMPT-?
294 001242 004767 000000G    JSR      PC,ERR3          ;YES, MISSING-OPERAND
295 001246 000167 000072    JMP      LD1X
296 001252 012767 000030 176372 40$: MOV      #T$BTAR,TD$TAR ;SELECT-BYTE-TRANS-REG
297 001260 016767 000000G 176364 MOV      MSTR2,TD$TDW ;MOVE-ADDR-TO-TRANSFER-REG
298 001266 012767 000020 176362 MOV      #T$BT,TD$TAW ;SELECT-BYTE-TRANS-MEMORY
299 001274 016767 000000G 176364 MOV      DATA1,TD$TDW ;LOAD-BYTE-TRANS
300 001302 066767 000000G 000000G ADD      INCVAL,MSTR2 ;ADVANCE-ADDRESS
301 001310 026767 000000G 000000G CMP      MSTR2,MEND ;HAS-UPPER-MEMORY-LIMIT-BEEN-REACHED
302 001316 101755          BLOS     40$          ;NO, CONTINUE
303 001320 032767 000000G 000000G BIT      #LOOP,BASE ;REPEAT-COMMAND
304 001326 001002          BNE      50$          ;YES, CONTINUE
305 001330 000167 000010    JMP      LD1X          ;ELSE-EXIT
306 001334 016767 000000G 000000G 50$: MOV      MSTR2,MSTR2 ;REINITIALIZE-ADDRESS
307 001342 000743          BR      40$
308                          ;
309                          ; LD1X:
310 001344          BIC      #RP,BASE ;CLEAR-PROMPT-FLAG
311 001344 042767 000000G 000000G JSR      PC,KILL ;KILL-AST-(IF-THERE-WAS-ONE)
312 001352 004767 000000G          JMP      COMXX
313 001356 000167 000000G

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

372.      ;
373.      ;
374.      ;
375.      ;
376.      ;
377.      ;
378 001510 004767 000000G.      JSR  PC,FIND.      ;SCAN COMMAND LINE
379 001514 103004      BCC  5$      ;SOMETHING THERE
380 001516 016767 000000G-000000G.  MOV  MSTRT,MEND.  ;SET END ADDR = START ADDR
381 001524 000445      BR   9$      ;JUMP TO RTN
382.      ;
383 001526 122711 000114      5$:  CMPB  #L,(R1)      ;LOOP INDICATOR
384 001532 001011      BNE  6$      ;NO. MUST BE UPPER ADDRESS
385 001534 052767 000000G-000000G.  BIS  #OUT,BASE.  ;SET FLAG FOR OUTPUT CONTROL
386 001542 016767 000000G-000000G.  MOV  MSTRT,MEND.  ;LOWER ADDRESS BECOMES UPPER ADDRESS
387 001550 004767 000000G.      JSR  PC,HANG.      ;HOW TO STOP LOOP
388 001554 000431      BR   9$      ;JUMP TO RTN
389.      ;
390 001556 004767 000000G.      6$:  JSR  PC,PACK.      ;CONVERT UPPER ADDRESS
391 001562 103004      BCC  7$      ;OK, CONTINUE
392 001564 004767 000000G.      JSR  PC,ERR4      ;INVALID NUMERIC VALUE
393 001570 000167 177550      JMP  LDIX      ;AND EXIT
394.      ;
395.      ;
396.      ;
397.      ;
398 001574 016767 000000G-000000G. 7$:  MOV  BINWD,MEND.  ;SET UP ENDING ADDRESS
399 001602 004767 000000G.      JSR  PC,FIND.      ;CHECK FOR LOOP INDICATOR
400 001606 103414      BCS  9$      ;NO LOOP
401 001610 122711 000114      CMPB  #L,(R1)      ;CORRECT LOOP INDICATOR
402 001614 001404      BEQ  8$      ;YES, CONTINUE
403 001616 004767 000000G.      JSR  PC,ERR1
404 001622 000167 000500      JMP  PR1X
405 001626 052767 000000G-000000G. 8$:  BIS  #OUT,BASE.  ;SET OUTPUT CONTROL
406 001634 004767 000000G.      JSR  PC,HANG.      ;HOW TO STOP LOOP
407.      ;
408 001640 016701 000000G.      9$:  MOV  RTNPT,R1      ;POINT TO ROUTINE
409 001644 000171 000000G.      JMP  @R1      ;JUMP TO ROUTINE
410.      ;
411.      ;
412.      ;
413.      ;
414 001650      PRAM:
415 001650 012767 000000G-000000G.  MOV  #T$FSA,A,PLAC. ;SELECT FSA A FOR SUBRTN
416 001656 016746 000000G.      MOV  AHIGH,-(SP)      ;UPPER MEMORY LIMIT
417 001662 016746 000000G.      MOV  ALOW,-(SP)      ;LOWER LIMIT
418 001666 004767 000000G.      JSR  PC,BUFFSA.  ;CHECK ADDRESS AGAINST MEMORY LIMITS
419 001672 103402      BCS  1$      ;CHECK FAILED
420 001674 004767 000000G.      JSR  PC,PRCNTL.  ;PRINT MEMORY
421 001700 000167 000422      1$:  JMP  PR1X      ;AND EXIT
422.      ;
423.      ;
424.      ;
425 001704      PRBM:
426 001704 012767 000004G-000000G.  MOV  #T$FSB,A,PLAC. ;SELECT FSA B FOR SUBRTN
427 001712 016746 000000G.      MOV  BHIGH,-(SP)      ;UPPER MEMORY LIMIT
428 001716 016746 000000G.      MOV  BLOW,-(SP)      ;LOWER LIMIT

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

429 001722 004767 000000G JSR PC,BUFFSA :CHECK ADDRESS AGAINST MEMORY LIMITS.
430 001726 103402 BCS 1$ :CHECK FAILED.
431 001730 004767 000000G JSR PC,PRCNTL :PRINT MEMORY.
432 001734 000167 000366 1$ JMP PR1X :AND EXIT.
433
434
435
436 001740 PRINT FROM FSA C.
437 001740 012767 000014 000000G PRCM:: MOV #T$FSA,APLACE :SELECT FSA FOR SUBRTN.
438 001746 016746 000000G MOV CHIGH,-(SP) :MOVE UPPER MEMORY LIMIT.
439 001752 016746 000000G MOV CLOW,-(SP) :LOWER LIMIT.
440 001756 004767 000000G JSR PC,BUFFSA :CHECK ADDRESS AGAINST MEMORY LIMITS.
441 001762 103402 BCS 1$ :CHECK FAILED.
442 001764 004767 000000G JSR PC,PRCNTL :PRINT MEMORY.
443 001770 000167 000332 1$ JMP PR1X :AND EXIT.
444
445
446
447
448 001774 PRINT FROM INPUT BUFFER.
449 001774 012767 000200 000000G PRIB:: MOV #T$DISK,APLACE :SELECT DISK SIMULATION.
450 002002 016746 000000G MOV IHIGH,-(SP) :SUPPLY UPPER MEMORY LIMIT.
451 002006 016746 000000G MOV ILOW,-(SP) :LOWER LIMIT.
452 002012 004767 000000G JSR PC,BUFFS2 :PREPARE TO LOAD.
453 002016 103543 BCS 1$ :ERROR ENCOUNTERED.
454
455 002020 016702 000000G 1$ MOV MSTR2,R2 :LOAD IB ADDRESS.
456 002024 011267 000000G MOV (R2),DATA1 :READ IB.
457 002030 004767 000000G JSR PC,PRDATA :CALL PRINT SUBROUTINE.
458
459 002034 066767 000000G 000000G ADD INCVAL,MSTR2 :ADVANCE ADDRESS.
460 002042 026767 000000G 000000G CMP MSTR2,MEND :HAS UPPER MEMORY LIMIT BEEN REACHED.
461 002050 101763 BLOS 1$ :NO, CONTINUE.
462 002052 032767 000000G 000000G BIT #LOOP,BASE :LOOP ON ?
463 002060 001522 BEQ PR1X :NO, EXIT.
464 002062 016767 000000G 000000G MOV MSTRT,MSTR2 :REINIT ADDRESS.
465 002070 000753 BR 1$ :START OVER.
466
467
468
469
470 002072 PRINT FROM OUTPUT BUFFER.
471 002072 012767 000040 000000G PROB:: MOV #T$ICD,APLACE :SELECT INTERFACE AND CONTROL FOR SUBRTN.
472 002100 016746 000000G MOV OHIGH,-(SP) :SUPPLY UPPER MEMORY LIMIT.
473 002104 016746 000000G MOV OLOW,-(SP) :LOWER LIMIT.
474 002110 004767 000000G JSR PC,BUFFS2 :PREPARE FOR LOAD.
475 002114 103504 BCS 1$ :ERROR.
476
477 002116 012767 000034 176362 1$ MOV #T$OBRA,TD$TAW :SELECT OUTPUT BUFFER WRITE REG.
478 002124 016767 000000G 176364 MOV MSTR2,TD$TDW :MOVE ADDR TO TRANSFER REG.
479 002132 012767 000036 176372 MOV #T$OB,TD$TAR :PREPARE FOR LCAD.
480 002140 016767 176374 000000G MOV TD$TDR,DATA1 :LOAD.
481 002146 004767 000000G JSR PC,PRDATA :CALL PRINT SUBROUTINE.
482
483 002152 066767 000000G 000000G ADD INCVAL,MSTR2 :ADVANCE ADDRESS.
484 002160 026767 000000G 000000G CMP MSTR2,MEND :HAS UPPER MEMORY LIMIT BEEN REACHED.
485 002166 101753 BLOS 1$ :NO, CONTINUE.

```

```

486 002170 032767 000000G-000000G- BIT- #LOOP,BASE- ;LOOP-ON ?-
487 002176 001453 BEQ- PR1X ;NO, EXIT-
488 002200 016767 000000G-000000G- MOV- MSTR1,MSTR2- ;REINIT-ADDRESS-
489 002205 000743 BR 1$ ;START-OVER-
490
491
492
493
494 002210 PRBT::
495 002210 012767 000040 000000G- MOV- #T#ICD,APLACE- ;SELECT-INTERFACE-AND- CNL FOR-SUBRTN-
496 002216 016746 000000G- MOV- THIGH,-(SP) ;SUPPLY-MEMORY-UPPER-LIMIT-
497 002222 016746 000000G- MOV- TLOW,-(SP) ;LOWER-LIMIT-
498 002226 004767 000000G- JSR PC,BUFSET- ;PREPARE FOR-LOAD-
499 002232 103435 BCS- PR1X ;ERROR-
500
501 002234 012767 000030 176362- 1$: MOV- #T#BTAR,TD$TAW ;SELECT-BYTE-TRANS-REG-
502 002242 016767 000000G-176364 MOV- MSTR2,TD$TDW ;MOVE-ADDR-TO-TRANSFER-REG-
503 002250 012767 000020 176372- MOV- #T#BT,TD$TAR- ;SELECT-BYTE-TRANS-MEMORY-
504 002256 016767 176374 000000G- MOV- TD$TDR,DATA1 ;LOAD-BYTE-TRANS-
505 002264 004767 000000G- JSR PC,PRDATA- ;CALL PRINT-SUBROUTINE-
506
507 002270 066767 000000G-000000G- ADD- INCVAL,MSTR2- ;ADVANCE-ADDRESS-
508 002276 026767 000000G-000000G- CMP- MSTR2,MEND ;HAS-UPPER-MEMORY-LIMIT-BEEN-REACHED-
509 002304 101753 BLOS- 1$ ;NO, CONTINUE-
510 002306 032767 000000G-000000G- BIT- #LOOP,BASE- ;LOOP-ON ?-
511 002314 001404 BEQ- PR1X ;NO, EXIT-
512 002316 016767 000000G-000000G- MOV- MSTR1,MSTR2- ;REINIT-ADDRESS-
513 002324 000743 BR 1$ ;START-OVER-
514
515
516 002326 PR1X:
517 002326 042767 000000C-000000G- BIC- #<OUT+ONCE>,BASE- ;CLEAR-CONTROL-FLAGS-
518 002334 004767 000000G- JSR PC,KILL- ;KILL-AST- (IF-THERE-WAS-ONE)
519 002340 000167 000000G- JMP COMXX-
520
521 000001 .END-

```

ALHIGH = ***** GX.	BYTE33 = 000041	BYTE85 = 000125	LD1 = 000060RG	002-Q#CHB = 000400
ALOW = ***** GX.	BYTE34 = 000042	BYTE86 = 000126	LD1LN = 000006 G	Q#CHRL = 000200
ALUCKE = 040000	BYTE35 = 000043	BYTE87 = 000127	LD1TBL = 000000RG	002-Q#CLR = 000040
ALUDE = 004000	BYTE36 = 000044	BYTE88 = 000130	LD1X = 001344R	002-Q#CNC = 030000
APLACE = ***** GX.	BYTE37 = 000045	BYTE89 = 000131	LOC.EN = 000100	Q#CP = 000060
A01 = 010000	BYTE38 = 000046	BYTE9 = 000011	LOC.WA = 040000	Q#CPCC = 000010
BASE = ***** GX.	BYTE39 = 000047	BYTE90 = 000132	LOC.WB = 100000	Q#CP2 = 000260
BHIGH = ***** GX.	BYTE4 = 000004	BYTE91 = 000133	LOOP = ***** GX.	Q#CSC = 010000
BINWD = ***** GX.	BYTE40 = 000050	BYTE92 = 000134	MAREN1 = 000001	Q#CSEL = 000360
BITVAL = 000000	BYTE41 = 000051	BYTE93 = 000135	MAREN2 = 004000	Q#CSET = 000002
BIT0 = 000001	BYTE42 = 000052	BYTE94 = 000136	MARLOD = 010000	Q#CSP = 020000
BIT1 = 000002	BYTE43 = 000053	BYTE95 = 000137	MAROUT = 000002	Q#DMA = 000001
BIT10 = 002000	BYTE44 = 000054	BYTE96 = 000140	MAR.LO = 002000	Q#ENBK = 040000
BIT11 = 004000	BYTE45 = 000055	BYTE97 = 000141	MAR.OU = 000040	Q#ENOP = 020000
BIT12 = 010000	BYTE46 = 000056	BYTE98 = 000142	MBKALL = 001000	Q#FAL = 004000
BIT13 = 020000	BYTE47 = 000057	BYTE99 = 000143	MBKCLK = 000400	Q#FC = 000045
BIT14 = 040000	BYTE48 = 000060	BYTVAL = 000144	MEND = ***** GX.	Q#FO = 000044
BIT15 = 100000	BYTE49 = 000061	CBKALL = 001000	MMADRD = 000100	Q#FP = 000046
BIT2 = 000004	BYTE5 = 000005	CBKCLK = 000400	MMLEFT = 000002	Q#HBF = 000002
BIT3 = 000010	BYTE50 = 000062	CHIGH = ***** GX.	MMOE = 000004	Q#ICP = 000006
BIT4 = 000020	BYTE51 = 000063	CLOW = ***** GX.	MMWRTE = 000010	Q#IHB = 000003
BIT5 = 000040	BYTE52 = 000064	CNOBRE = 100000	MNOBRE = 100000	Q#IHLR = 000002
BIT6 = 000100	BYTE53 = 000065	COMXX = ***** GX.	MREN1 = 000001	Q#IMRP = 000007
BIT7 = 000200	BYTE54 = 000066	CPCCEN = 010000	MREN2 = 020000	Q#LBD = 001000
BIT8 = 000400	BYTE55 = 000067	CPREAD = 040000	MSRT = ***** GX.	Q#LBDP = 001001
BIT9 = 001000	BYTE56 = 000070	CPWRTE = 020000	MSTR2 = ***** GX.	Q#LBP = 000001
BLOW = ***** GX.	BYTE57 = 000071	CSADDR = 000004	MSYN = 000040	Q#LDCD = 000003
BUFFSA = ***** GX.	BYTE58 = 000072	CSEDCI = 100000	N = 000144	Q#LDM = 000004
BUFFSET = ***** GX.	BYTE59 = 000073	CSOE = 000040	OHIGH = ***** GX.	Q#LDPP = 002000
BUFF2 = ***** GX.	BYTE6 = 000006	CSWRTE = 000100	OLOW = ***** GX.	Q#LHP = 010000
BYTE0 = 000000	BYTE60 = 000074	DATA1 = ***** GX.	ONCE = ***** GX.	Q#MNC = 114000
BYTE1 = 000001	BYTE61 = 000075	DBR.RD = 000001	OUT = ***** GX.	Q#MR = 000052
BYTE10 = 000012	BYTE62 = 000076	DB\$CPP = 001457	PACK = ***** GX.	Q#MRP = 000040
BYTE11 = 000013	BYTE63 = 000077	DB\$SPT = 000026	PDATA = ***** GX.	Q#MRP2 = 000240
BYTE12 = 000014	BYTE64 = 000100	DB\$TPC = 000023	PLB = 000010	Q#MSC = 040000
BYTE13 = 000015	BYTE65 = 000101	DISPGS = 100000	PLC = 000020	Q#MSET = 000004
BYTE14 = 000016	BYTE66 = 000102	DMAAUR = 000005	PLD = 000030	Q#MSP = 100000
BYTE15 = 000017	BYTE67 = 000103	DMARRD = 000003	PLRWR = 000200	Q#NCLK = 176000
BYTE16 = 000020	BYTE68 = 000104	DMARRW = 000004	PLR.EN = 000200	Q#PP = 000100
BYTE17 = 000021	BYTE69 = 000105	ENBR = 010000	PRAM = 001650RG	002-Q#PPSW = 000320
BYTE18 = 000022	BYTE7 = 000007	ERR1 = ***** GX.	PRBM = 001704RG	002-Q#PP2 = 000300
BYTE19 = 000023	BYTE70 = 000106	ERR3 = ***** GX.	PRBT = 002210RG	002-Q#QLT = 000013
BYTE2 = 000002	BYTE71 = 000107	ERR4 = ***** GX.	PRCM = 001740RG	002-Q#QL = 000043
BYTE20 = 000024	BYTE72 = 000110	ERR6 = ***** GX.	PRCNTL = ***** GX.	Q#QLA = 000053
BYTE21 = 000025	BYTE73 = 000111	FIND = ***** GX.	PRDATA = ***** GX.	Q#QLB = 000054
BYTE22 = 000026	BYTE74 = 000112	HANG = ***** GX.	PRIB = 001774RG	002-Q#QLR = 000001
BYTE23 = 000027	BYTE75 = 000113	IHIGH = ***** GX.	PROB = 002072RG	002-Q#QW = 000042
BYTE24 = 000030	BYTE76 = 000114	ILOW = ***** GX.	PR1 = 001362RG	002-Q#RDCD = 000005
BYTE25 = 000031	BYTE77 = 000115	INCVL = ***** GX.	PR1LN = 000006 G	Q#RDM = 000006
BYTE26 = 000032	BYTE78 = 000116	KILL = ***** GX.	PR1TBL = 000030RG	002-Q#REBK = 001000
BYTE27 = 000033	BYTE79 = 000117	LDAM = 000332RG	002-PR1X = 002326RG	002-Q#RNC = 006000
BYTE28 = 000034	BYTE8 = 000010	LDBM = 000366RG	002-Q#SCR1 = 176420	Q#RSC = 004000
BYTE29 = 000035	BYTE80 = 000120	LDBT = 001112RG	002-Q#SCR2 = 176422	Q#RSET = 000010
BYTE3 = 000003	BYTE81 = 000121	LDCM = 000422RG	002-Q#SLBR = 176424	Q#SM = 100000
BYTE30 = 000036	BYTE82 = 000122	LDCNTL = ***** GX.	Q#ATTN = 000100	Q#SP = 000120
BYTE31 = 000037	BYTE83 = 000123	LDIB = 000456RG	002-Q#BCL = 000001	Q#SP2 = 000340
BYTE32 = 000040	BYTE84 = 000124	LDIB = 000654RG	002-Q#CCCP = 000040	RQO.EN = 000200

TMEM...MACRO.M1110 27-MAR-80 15:37 PAGE 6-5
SYMBOL TABLE

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

RG0.VA=000000	T\$CLK=0002000	WORD1=0000002	WORD40=000120	WORD71=000216
RP=00000000 GX	T\$DISK=0002000	WORD10=0000024	WORD41=000122	WORD72=000220
RTNPT=00000000 GX	T\$DRD=0000004	WORD11=0000026	WORD42=000124	WORD73=000222
SCAN=00000000 GX	T\$EMEM=0100000	WORD12=0000030	WORD43=000126	WORD74=000224
SEQ.CI=0000010	T\$FSAA=0000000	WORD13=0000032	WORD44=000130	WORD75=000226
S\$CLR=0000000	T\$FSAB=0000004	WORD14=0000034	WORD45=000132	WORD76=000230
S\$LA=0000001	T\$FSAC=0000014	WORD15=0000036	WORD46=000134	WORD77=000232
S\$OB=0000005	T\$FSB2=0000010	WORD16=0000040	WORD47=000136	WORD78=000234
S\$OR=0000006	T\$IB=0000026	WORD17=0000042	WORD48=000140	WORD79=000236
S\$OX=0000004	T\$IBAR=0000024	WORD18=0000044	WORD49=000142	WORD80=000240
S\$SR=0000007	T\$IBE=0200000	WORD19=0000046	WORD50=000144	WORD81=000242
S\$S1=0000010	T\$IBF=0400000	WORD20=0000050	WORD51=000146	WORD82=000244
S\$S2=0000014	T\$ICD=0000040	WORD21=0000052	WORD52=000150	WORD83=000246
TD\$CTR=176370	T\$MODE=0040000	WORD22=0000054	WORD53=000152	WORD84=000250
TD\$CTW=176360	T\$OB=0000036	WORD23=0000056	WORD54=000154	WORD85=000252
TD\$INL=0040000	T\$OBE=0040000	WORD24=0000060	WORD55=000156	WORD86=000254
TD\$MEM=000270	T\$OBF=0100000	WORD25=0000062	WORD56=000160	WORD87=000256
TD\$OAR=176344	T\$OBRA=0000034	WORD26=0000064	WORD57=000162	WORD88=000260
TD\$OTR=176346	T\$OBWA=0000032	WORD27=0000066	WORD58=000164	WORD89=000262
TD\$ORD=000274	T\$OUTH=1000000	WORD28=0000070	WORD59=000166	WORD90=000264
TD\$SW=176376	T\$RBD0=0002000	WORD29=0000072	WORD60=000170	WORD91=000266
TD\$TAR=176372	T\$RNB=0000040	WORD30=0000074	WORD61=000172	WORD92=000270
TD\$TAW=176362	T\$RSET=0400000	WORD31=0000076	WORD62=000174	WORD93=000272
TD\$TDR=176374	T\$SC=0000022	WORD32=0000080	WORD63=000176	WORD94=000274
TD\$TDW=176364	T\$SCLK=0200000	WORD33=0000082	WORD64=000200	WORD95=000276
THIGH=00000000 GX	T\$SEG1=0000000	WORD34=0000084	WORD65=000202	WORD96=000300
TLOW=00000000 GX	T\$SEG2=0000001	WORD35=0000086	WORD66=000204	WORD97=000302
T\$AD=0000020	T\$SEG3=0000002	WORD36=0000088	WORD67=000206	WORD98=000304
T\$BA=0000002	T\$SO=0000001	WORD37=0000090	WORD68=000210	WORD99=000306
T\$BD=0000010	T\$UBUS=1000000	WORD38=0000092	WORD69=000212	WORD100=000310
T\$BSO=1000000	T\$1CLK=0004000	WORD39=0000094	WORD70=000214	XTREAD=001000
T\$BT=0000020	T\$BBEN=0000020	WORD40=0000096		XTWRITE=000400
T\$BTAR=000030	UBD.IN=0000020			
T\$BTD=0002000	WORD0=0000000			
T\$CD=000100				

. ABS. 000000 000
000000 001
TMEM. 002344 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3537 WORDS (14 PAGES)
DYNAMIC MEMORY: 4916 WORDS (18 PAGES)
ELAPSED TIME: 00:00:51
TMEM, TMEM/SP=C20,1 JIM,C20,1 JTMEM

```

1      .TITLE..TDATA.
2      .PSECT..TDATA.
3      .LIST..MEB.
4
5      ;
6      ;
7      ;
8      ;
9      ;
10     ;
11     ;
12     ;
13     ;
14     ;
15     ;
16     ;
17     ;
18     ;
19     ;
20     ;
21     ;
22     ;
23     ;
24     ;
25     ;
26     ;
27     ;
28     ;
29     ;
30     ;
31     ;
32     ;
33     000000      .CH1TBL::
34     000000      .ASCII.. /IB/      ; INPUT BUFFER.
35     000002      .WORD.. CH1B      ;
36     000004      .ASCII.. /BT/      ; BYTE TRANSLATOR.
37     000006      .WORD.. CHBT      ;
38     000002      CH1LN.. == <.-CH1TBL>/4
39
40     ;
41     ;
42     000010      .LIST.
43     000010      .LI1TBL::
44     000012      .ASCII.. /IB/      ; INPUT BUFFER.
45     000014      .WORD.. LI1B      ;
46     000016      .ASCII.. /BT/      ; BYTE TRANSLATOR.
47     000002      .WORD.. LI1T      ;
48     000002      LI1LN.. == <.-LI1TBL>/4
49
50     ;
51     ;
52     000020      .DATA FILES.
53     000020      .DF1TBL::
54     000022      .ASCII.. /IB/      ; INPUT BUFFER.
55     000024      .WORD.. DF1B      ;
56     000026      .ASCII.. /BT/      ; BYTE TRANSLATOR.
57     000002      .WORD.. DF1T      ;
58     000002      DF1LN.. == <.-DF1TBL>/4

```

```

58      ;      TRANSFER
59      ;
60      ;      TR1TBL::
61      000030      111      102      .ASCII /IB/      ; INPUT BUFFER
62      000032      002134      .WORD TRIB
63      000034      102      .ASCII /BT/      ; BYTE TRANSLATOR
64      000036      002226      .WORD TRBT
65      000002      TR1LN == <.-TR1TBL>/4
66      ;
67      ;
68      ;      DATA
69      ;
70      ;
71      ;
72      ;      NO OPERANDS.
73      ;      THIS ROUTINE PUTS OUT PROMPTS FOR DATA AND READS THE COMMAND
74      ;      LINE. IT THEN CALLS A SUBROUTINE IN TMAIN WHICH PROCESSES
75      ;      THE COMMAND LINE AND PLACES IT INTO TWO BUFFERS. ONE OF THE
76      ;      BUFFERS IS SIMPLY A MIRROR OF THE COMMAND LINE INPUT TO BE
77      ;      USED FOR VERIFICATION. THE OTHER BUFFER IS THE 6-BIT BYTE
78      ;      BUFFER WHICH WILL EVENTUALLY BE TRANSFERRED TO THE
79      ;      TERM DETECTOR INPUT BUFFER (SEE THE 'TR' AND 'RU'
80      ;      COMMANDS).
81      ;
82      ;      NB. SINCE THE LOCAL INPUT BUFFER IS TRANSFERRED TO
83      ;      THE TERM DETECTOR IN 16-BIT WORDS AND THERE ARE 8
84      ;      6-BIT BYTES IN 3 16-BIT WORDS, THE NUMBER OF 6-BIT
85      ;      BYTES IN THE LOCAL DATA BUFFER SHOULD BE DIVISIBLE
86      ;      BY 8. NO CHECKING IS DONE, HOWEVER.
87      ;
88      ;      CLEAR LOCAL DATA BUFFER (BOTH THE 6-BIT BYTE TABLE
89      ;      AND THE 8-BIT ASCII MIRROR TABLE).
90      ;
91      000040      DA1::
92      000040      005067      000000G      CLR COUNT      ; CLEAR BIT SHIFT COUNTER
93      000044      005067      000000G      CLR DATALN      ; CLEAR LENGTH OF 'DATA'
94      000050      012700      000000G      MOV #DTBL,R0      ; POINT TO 6-BIT BYTE TABLE
95      000054      012701      001000      MOV #<<1364,*3>/8,+1,R1 ; NUMBER OF WORDS IN TABLE
96      000060      005020      1$: CLR (R0)+      ; CLEAR DATA TABLE
97      000062      005301      DEC R1
98      000064      001375      BNE 1$
99      ;
100     000066      012700      000000G      MOV #DSAVE,R0      ; POINT TO ASCII MIRROR TABLE
101     000072      012701      001252      MOV #1364/2,R1      ; NUMBER OF WORDS
102     000076      005020      2$: CLR (R0)+      ; CLEAR TABLE
103     000100      005301      DEC R1
104     000102      003375      BGT 2$
105     ;
106     ;
107     ;      READ 'DATA' FROM CONSOLE.
108     ;      CALL ROUTINE TO ADD 1 COMMAND LINE'S WORTH OF DATA
109     ;      TO THE TABLES. A <CR> RESPONSE TO THE PROMPT
110     ;      TERMINATES THE COMMAND.
111     000104      012704      000000G      MOV #DTBL,R4      ; R4 -> 6-BIT BYTE TABLE
112     000110      012705      000000G      MOV #DSAVE,R5      ; R5 -> MIRROR INPUT TABLE
113     000114      004767      000000G      DATA: JSR PC,GCONLY      ; GET COMMAND LINE
114     000120      005767      000000G      TST GCMLN      ; ANYTHING THERE

```

TDATA: MACRO:M1110 27-MAR-80 15:36 PAGE:5-2.

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

115	000124	001407		BEQ.	DA1X	;NO. EXIT.
116	000126	004767	000000G.	JSR.	PC,DBLD.	;ELSE PROCESS DATA CHARS.
117	000132	103404		BCS.	DA1X	;DATA OVERFLOW OR ERROR.
118	000134	066767	000000G.000000G.	AID.	DPLUS,DATALN.	;ACCUMULATE LENGTH OF DATA.
119	000142	000764		BR	DATA	;GET ANOTHER LINE.
120						
121	000144			DA1X:		
122	000144	000167	000000G.	JMP.	COMXX.	


```
124      ;
125      ;
126      ;
127      ;
128      ;
129      ;
130      ;
131      ;
132      ;
133 000150      CH1::
134 000150 004767 000000G  JSR  PC,FIND  ;FIND THE BUFFER MNEMONIC IN COMMAND LINE
135 000154 103004      BCC  1$      ;OK, CONTINUE
136 000156 004767 000000G  JSR  PC,ERR3  ;MISSING OPERAND
137 000162 000167 000360  JMP  CH1X
138      ;
139      ;
140      ;
141      ;
142 000166 012700 000002  1$:  MOV  #CH1LN,R0  ;NUMBER OF TABLE ENTRIES
143 000172 012702 000000*  MOV  #CH1TBL,R2 ;TABLE OF RESPONSES/RTN ADDRESSES
144 000176 004767 000000G  JSR  PC,SCAN  ;CHECK COMMAND LINE AGAINST TABLE
145 000202 103004      BCC  2$      ;OK, CONTINUE
146 000204 004767 000000G  JSR  PC,ERR6  ;INVALID MEMORY MNEMONIC
147 000210 000167 000332  JMP  CH1X
148      ;
149      ;
150      ;
151      ;
152      ;
153      ;
154 000214 010167 000000G  2$:  MOV  R1,RTNPT  ;SAVE POINTER TO RTN ADDRESS
155 000220 004767 000000G  JSR  PC,FIND  ;LOCATE START CHANGE ADDR IN COMMAND LINE
156 000224 103003      BCC  3$      ;OK, CONTINUE
157 000226 004767 000000G  JSR  PC,ERR3  ;MISSING OPERAND
158 000232 000545      BR   CH1X  ;EXIT
159 000234 004767 000000G  3$:  JSR  PC,PACK  ;CONVERT COMMAND LINE VALUE TO BINARY
160 000240 103003      BCC  4$      ;CONVERSION SUCCESSFUL
161 000242 004767 000000G  JSR  PC,ERR4  ;INVALID NUMERIC VALUE
162 000246 000537      BR   CH1X
163      ;
164      ;
165      ;
166 000250 016701 000000G  4$:  MOV  RTNPT,R1  ;POINT TO ROUTINE
167 000254 000171 000000  JMP  @R1      ;GO THERE
168      ;
169      ;
170      ;
171      ;
172      ;
173      ;
174      ;
175      ;
176      ;
177      ;
178      ;
179 000260      CHIB::
180 000260 005767 000000G  TST  DATALN  ;HAS 'DATA' COMMAND BEEN EXECUTED
```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

181 000264 001003      BNE 40$      ;YES, CONTINUE.
182 000266 004767 000000G JSR PC,ERR7    ;DATA MUST PRECEED CHANGE.
183 000272 000525      BR CH1X
184 ;
185 ; DETERMINE WHERE IN THE TWO TABLES (6-BIT BYTE AND 8-BIT
186 ; ASCII MIRROR TABLE) TO START CHANGING. DO THIS FOR THE
187 ; 6-BIT BYTE TABLE BY CONVERTING THE START CHANGE ADDRESS
188 ; FROM THE COMMAND LINE INTO A AN OFFSET INTO THE 6-BIT
189 ; BYTE TABLE. THE FIELD 'OFF6' IS THE OUTPUT FROM THE
190 ; ROUTINE OFFSET. THE FIELD 'BINWD' IS THE START CHANGE
191 ; ADDRESS FROM THE COMMAND LINE (IN BINARY).
192 ;
193 000274 004767 000000G 40$: JSR PC,OFFSET    ;CALCULATE 6-BIT TABLE OFFSET.
194 000300 012704 000000G MOV #DTBL,R4      ;R4 -> 6-BIT TABLE
195 000304 066704 000000G ADD OFF6,R4      ;START CHANGING HERE.
196 000310 012705 000000G MOV #DSAVE,R5     ;R5 -> 8-BIT MIRROR TABLE.
197 000314 066705 000000G ADD BINWD,R5      ;START CHANGING HERE.
198 000320 016767 000000G-000000G MOV BINWD,CHLEN ;INIT FOR NEW LENGTH OF BUFFER.
199 ;
200 ; PUT OUT PROMPT FOR CHANGE DATA. CALL SUBROUTINE TO
201 ; PROCESS COMMAND LINE (SEE 'DA' COMMAND ABOVE).
202 ; ADD THE NUMBER OF 6-BIT CHARACTERS FROM THE
203 ; PROCESSED COMMAND LINE TO THE CHANGE START
204 ; ADDRESS.
205 ;
206 000326 004767 000000G 5$: JSR PC,GCONLY    ;GET COMMAND LINE.
207 000332 005767 000000G TST GCMLN      ;ANYTHING THERE.
208 000336 001407      BEQ 6$      ;NO. <CR> RESPONSE
209 000340 004767 000000G JSR PC,DBLD    ;CHANGE 'DATA' TABLE.
210 000344 103500      BCS CH1X      ;ERROR, EXIT.
211 000346 066767 000000G-000000G ADD DPLUS,CHLEN ;ACCUMULATE LENGTH OF DATA BUFFER.
212 000354 000764      BR 5$      ;PROMPT FOR MORE CHANGE DATA.
213 ;
214 ; SPECIAL PROCESSING:
215 ; THE FIELD 'DATALN' CONTAINS THE NUMBER OF 6-BIT BYTES
216 ; IN THE LOCAL DATA BUFFER (SEE 'DA' COMMAND). IF THE
217 ; CHANGE COMMAND IS IN THE FORM:
218 ; >CH IB 2.
219 ; AND THE DATA BUFFER CONTAINS 'ABCDE' AND THE CHANGE DATA IS
220 ; '1234', THE DATA BUFFER WILL CONTAIN 'AB1234' AND DATALN WILL
221 ; BE UPDATED TO 6. IF, HOWEVER, THE CHANGE DATA IS '12' THEN
222 ; THE DATA BUFFER WILL CONTAIN 'AB12D' AND DATALN WILL REMAIN
223 ; AT 5. THE FIELD 'DATALN' CONTAINS THE LENGTH OF THE DATA
224 ; BUFFER BEFORE ANY CHANGES. THE FIELD 'CHLEN' CONTAINS
225 ; THE LENGTH OF THE BUFFER AFTER THE CHANGES. THE LARGER
226 ; OF THE TWO WILL BECOME THE NEW 'DATALN'.
227 ;
228 000356 026767 000000G-000000G 6$: CMP CHLEN,DATALN ;WHICH IS BIGGER.
229 000364 003470      BLE CH1X      ;DATALN.
230 000366 016767 000000G-000000G MOV CHLEN,DATALN ;REPLACE DATALN.
231 000374 000464      BR CH1X      ;EXIT.
232 ;
233 ;
234 ; BYTE TRANSLATOR.
235 ; THE FIELD 'BINWD' CONTAINS THE CHANGE START ADDRESS.
236 ; WHICH IS A ZERO RELATIVE WORD NUMBER. EG. THE COMMAND
237 ; >CH BT 7

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

238      ;      WILL START CHANGING THE BYTE TRANSLATOR OVERRIDE TABLE.
239      ;      (SEE BELOW) AT WORD 7.
240      ;
241      ;
242      000376      CHBT:
243      000376      026727      000000G-000000C      CMP      BINWD,*(BTEND-BTRANS) ;START ADDRESS IN RANGE.
244      000404      103403      BLO      3$      ;YES, CONTINUE.
245      000406      004767      JSR      PC,ERR20      ;START ADDRESS OUT OF RANGE HIGH
246      000412      000455      BR      CH1X      ;EXIT.
247      ;
248      ;      THE CHANGE COMMAND CREATES A BYTE TRANSLATOR OVERRIDE TABLE
249      ;      (WHICH IS A COMPLETE COPY OF THE BYTE TRANSLATOR TABLE). ALL
250      ;      ALL CHANGES ARE APPLIED TO THE OVERRIDE TABLE. THE OVERRIDE
251      ;      TABLE IS TRANSFERRED TO THE TERM DETECTOR BY EITHER THE 'TR'
252      ;      OR THE 'RU' COMMAND. IF THIS ROUTINE IS BEING ENTERED FOR
253      ;      THE FIRST TIME, COPY THE BYTE TRANSLATOR TABLE TO THE
254      ;      OVERRIDE TABLE.
255      ;
256      000414      032767      000000G-000000G-3$      BIT      *RIDE,BASE      ;FIRST USE OF 'CHANGE'
257      000422      001014      BNE      4$      ;NO, LEAVE OVERRIDE TABLE ALONE.
258      000424      052767      000000G-000000G      BIS      *RIDE,BASE      ;FLAG ROUTINE ENTERED.
259      000432      012700      000000G      MOV      *BTRANS,R0      ;POINT TO DEFAULT TABLE.
260      000436      012701      000000G      MOV      *BTOVER,R1      ;POINT TO OVERRIDE TABLE.
261      000442      012702      000000C      MOV      *(BTEND-BTRANS)/2,R2 ;NUMBER OF WORDS.
262      000446      012021      30$      MOV      (R0)+(R1)+      ;LOAD OVERRIDE TABLE.
263      000450      005302      DEC      R2
264      000452      001375      BNE      30$
265      ;
266      ;      LOAD THE CHANGE START ADDRESS IN BYTES INTO R2.
267      ;
268      000454      012702      000000G      4$      MOV      *BTOVER,R2      ;POINT TO OVERRIDE TABLE.
269      000460      016703      000000G      MOV      BINWD,R3      ;LOAD WORD OFFSET.
270      000464      006303      ASL      R3      ;SHIFT FOR BYTE OFFSET.
271      000466      060302      ADD      R3,R2      ;START CHANGING HERE.
272      ;
273      ;      PROMPT FOR A LINE OF CHANGE DATA. THE COMMAND LINE
274      ;      MAY CONTAIN MORE THAN ONE WORD OF DATA. EG:
275      ;      >000000 000000 000000 000000 000000 000000
276      ;      CHANGE 6 CONTIGUOUS WORDS WORTH OF BYTE TRANSLATOR CODES.
277      ;      A <CR> RESPONSE TO THE PROMPT TERMINATES THE COMMAND.
278      ;
279      000470      004767      000000G      5$      JSR      PC,GCONLY      ;READ COMMAND LINE
280      000474      004767      000000G      JSR      PC,FIND      ;LOCATE AN OVERRIDE WORD.
281      000500      103422      BCS      CH1X      ;<CR> RESPONSE, EXIT.
282      ;
283      ;      CONVERT THE OVERRIDE VALUE TO BINARY. ANY ERROR IN
284      ;      CONVERSION TERMINATES THE COMMAND.
285      ;
286      000502      004767      000000G      6$      JSR      PC,PACK      ;CONVERT NUMERIC VALUE.
287      000506      103003      BCC      8$      ;OK, CONTINUE.
288      000510      004767      000000G      JSR      PC,ERR4      ;INVALID NUMERIC VALUE.
289      000514      000414      BR      CH1X      ;EXIT.
290      ;
291      ;      CHECK FOR TABLE OVERFLOW. IF OVERFLOW, TERMINATE COMMAND.
292      ;
293      000516      020227      000000C      8$      CMP      R2,*BTOVER+(BTEND-BTRANS) ;END OF TABLE REACHED.
294      000522      103403      BLO      10$      ;NO, CONTINUE.

```

TDATA: MACRO:M1110 27-MAR-80 15:36 PAGE:6-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

295 000524 004767 000000G	JSR	PC,ERR17	;TABLE OVERFLOW
296 000530 000406	BR	CHIX	;EXIT
297 000532 016722 000000G	MOV	BINWD,(P2)+	;ENTER WORD IN BT TABLE
298 000536 004767 000000G	JSR	PC,FIND	;LOCATE NEXT OVERRIDE WORD
299 000542 183357	BCC	6\$;PROCESS STRING
300 000544 000751	BR	5\$;GET COMMAND LINE
301	:		
302	:		
303 000546	CHIX:		
304 000546 000167 000000G	JMP	COMXX	

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
306 ;
307 ;
308 ;
309 ;
310 ;
311 ;
312 ;
313 ;
314 ;
315 000552. ;
316 000552. 004767 000000G. ;
317 000556 103004 ;
318 000560 004767 000000G. ;
319 000564 000167 000424 ;
320 ;
321 ;
322 ;
323 ;
324 000570 012700 000002 1$: MOV. #LI1LN,R0 ;NUMBER OF TABLE ENTRIES.
325 000574 012702. 000010. MOV. #LI1TBL,R2. ;POINT TO TABLE.
326 000600 004767 000000G. JSR. PC,SCAN. ;MATCH AGAINST COMMAND LINE.
327 000604 103004 BCC. 2$ ;MATCH WAS MADE.
328 000606 004767 000000G. JSR. PC,ERR6 ;INVALID MEMORY MNEMONIC.
329 000612. 000167 000376 JMP. LI1X
330 ;
331 ;
332 ;
333 ;
334 ;
335 ;
336 000616 010167 000000G. 2$: MOV. R1,RTNPT. ;SAVE POINTER.
337 000622. 004767 000000G. JSR. PC,FIND. ;LOCATE LIST START ADDR IN COMMAND LINE.
338 000626 103004 BCC. 3$ ;OK, CONTINUE.
339 000630 004767 000000G. JSR. PC,ERR3 ;MISSING OPERAND.
340 000634 000167 000354 JMP. LI1X ;EXIT.
341 000640 004767 000000G. 3$: JSR. PC,PACK. ;CONVERT COMMAND LINE VALUE TO BINARY.
342 000644 103004 BCC. 4$ ;CONVERSION SUCCESSFUL.
343 000646 004767 000000G. JSR. PC,ERR4 ;INVALID NUMERIC VALUE.
344 000652. 000167 000336 JMP. LI1X
345 ;
346 ;
347 ;
348 000656 004767 000000G. 4$: JSR. PC,HANG. ;HOW TO STOP PRINT
349 000662. 016701 000000G. MOV. RTNPT,R1 ;POINT TO ROUTINE.
350 000666 000171 000000 JMP. @ (R1) ;JUMP TO ROUTINE.
351 ;
352 ;
353 ;
354 ;
355 ;
356 ;
357 ;
358 ;
359 000672. ;
360 000672. 005767 000000G. ;
361 000676 001003 ;
362 000700 004767 000000G. ;
```

LIST.
PERFORM SECOND LEVEL PARSING.
EG. IN THE COMMAND.
>LI1 IB 0
PARSE THE 'IB'

LI1::
JSR. PC,FIND. ;LOCATE BUFFER MNEMONIC IN COMMAND LINE.
BCC. 1\$;OK, CONTINUE.
JSR. PC,ERR3 ;MISSING OPERAND.
JMP. LI1X ;EXIT.

MATCH THE MNEMONIC FROM THE COMMAND LINE AGAINST THE
TABLE OF VALID MNEMONICS.

1\$: MOV. #LI1LN,R0 ;NUMBER OF TABLE ENTRIES.
MOV. #LI1TBL,R2. ;POINT TO TABLE.
JSR. PC,SCAN. ;MATCH AGAINST COMMAND LINE.
BCC. 2\$;MATCH WAS MADE.
JSR. PC,ERR6 ;INVALID MEMORY MNEMONIC.
JMP. LI1X

SAVE THE POINTER TO THE ROUTINE ASSOCIATED WITH THE
BUFFER, R1 -> ROUTINE ADDRESS.
SCAN THE COMMAND LINE FOR THE 'START LIST ADDRESS'.
THE START ADDRESS IS ZERO-RELATIVE.

2\$: MOV. R1,RTNPT. ;SAVE POINTER.
JSR. PC,FIND. ;LOCATE LIST START ADDR IN COMMAND LINE.
BCC. 3\$;OK, CONTINUE.
JSR. PC,ERR3 ;MISSING OPERAND.
JMP. LI1X ;EXIT.

3\$: JSR. PC,PACK. ;CONVERT COMMAND LINE VALUE TO BINARY.
BCC. 4\$;CONVERSION SUCCESSFUL.
JSR. PC,ERR4 ;INVALID NUMERIC VALUE.
JMP. LI1X

JUMP TO THE ROUTINE TO LIST BUFFER CONTENTS.

4\$: JSR. PC,HANG. ;HOW TO STOP PRINT
MOV. RTNPT,R1 ;POINT TO ROUTINE.
JMP. @ (R1) ;JUMP TO ROUTINE.

INPUT BUFFER

CHECK WHETHER THERE IS ANY 6-BIT BYTE DATA IN THE
TABLE.

LI1B::
TST. DATALN. ;HAS 'DATA' COMMAND BEEN EXECUTED.
BNE. 20\$;YES, CONTINUE.
JSR. PC,ERR8 ;DATA COMMAND MUST PRECEED LIST

```

363 000704 000543      BR      LI1X
364      :
365      :
366      :
367 000706 026767 000000G 000000G 20$: CMP  BINWD,DATALN  :IS LIST START ADDRESS IN RANGE
368 000714 002403      BLT  25$      :YES, CONTINUE
369 000716 004767 000000G      JSR  PC,ERR9      :OUT OF RANGE
370 000722 000534      BR      LI1X
371      :
372      :
373      :
374      :
375      :
376      :
377 000724 012700 000000G      25$: MOV  #DSAVE,R0      :POINT TO ASCII VERSION OF 'DATA'
378 000730 066700 000000G      ADD  BINWD,R0      :START LIST HERE
379 000734 005001      CLR  R1      :CLEAR PRINT LINE CHAR COUNTER
380 000736 012705 000000G      MOV  #PRINT,R5      :R5 -> PRINT LINE
381      :
382      :
383      :
384      :
385      :
386 000742 032767 000000G 000000G 3$: BIT  #LOOP,BASE      :FINISHED?
387 000750 001521      BEQ  LI1X      :YES, EXIT
388 000752 105710      TSTB (R0)      :END OF TABLE
389 000754 001003      BNE  30$      :NO, CONTINUE
390 000756 004767 000000G      JSR  PC,CONSOL      :PRINT LAST LINE
391 000762 000514      BR      LI1X      :AND EXIT
392      :
393      :
394      :
395      :
396 000764 122710 000015      30$: CMPB  #15,(R0)      :<CR> IN ASCII DATA
397 000770 002006      BGE  4$      :PROCESS <CR> OF <LF>
398 000772 005201      INC  R1      :NO, WRITE CURRENT LINE
399 000774 022701 000110      CMP  #72,R1      :ADD TO PRINT LINE CHAR COUNT
400 001000 002426      BLT  7$      :PRINT LINE FULL
401 001002 112025      MOVB (R0)+,(R5)+ :YES, WRITE TO TT0
402 001004 000756      BR      3$      :MOVE CHAR TO PRINT LINE
403      :
404      :
405      :
406 001006 062701 000004      4$: ADD  #4,R1      :GET NEXT CHAR
407 001012 022701 000110      CMP  #72,R1      :ADD FOR ANGLE BRACKETS
408 001016 002417      BLT  7$      :ENOUGH SPACE ON CURRENT LINE
409 001020 112725 000074      MOVB #1,(R5)+ :NO, WRITE CURRENT LINE
410 001024 112725 000061      MOVB #1,(R5)+ :MOVE LEFT BRACKET IN ANY CASE
411 001030 122720 000012      CMPB #12,(R0)+ :MOVE ASCII 1 IN ANY CASE
412 001034 001403      BEQ  5$      :LINE FEED
413 001036 112725 000065      MOVB #5,(R5)+ :YES, BUILD ASCII STRING
414 001042 000402      BR      6$      :FINISH WITH RIGHT BRACKET
415 001044 112725 000062      5$: MOVB #2,(R5)+
416 001050 112725 000076      6$: MOVB #2,(R5)+ :CLOSE BRACKETS
417 001054 000732      BR      3$      :GET NEXT CHAR
418      :
419      :
420      :
421 001056 004767 000000G      7$: JSR  PC,CONSOL      :WRITE LINE TO TT0

```

```

420 001062 005001 CLR R1 ;CLEAR PRINT LINE CHAR COUNT
421 001064 312705 MOV #PRINT,R5 ;RESET PRINT LINE POINTER
422 001070 000724 BR 3$ ;PRINT NEXT 78 CHARS
423 ;
424 ;
425 ;
426 ;
427 ;
428 ;
429 ;
430 ;
431 ;
432 001072 LIBT::
433 001072 032767 000000G-000000G BIT #RIDE,BASE ;HAS BT TABLE BEEN OVERRIDEN
434 001100 001003 BNE 1$ ;YES, CONTINUE
435 001102 004767 000000G JSR PC,ERR22 ;FOR BT, CHANGE COMMAND MUST PRECEED LIST
436 001106 000442 BR LI1X ;EXIT
437 ;
438 ;
439 ;
440 ;
441 ;
442 ;
443 001110 026727 000000G-000000G 1$: CMP BINWD,#(BTEND-BTRANS) ;IS START ADDRESS IN RANGE
444 001116 103403 BLO 2$ ;YES, CONTINUE
445 001120 004767 000000G JSR PC,ERR20 ;START ADDRESS OUT OF RANGE HIGH
446 001124 000433 BR LI1X ;EXIT
447 ;
448 ;
449 ;
450 001126 012702 000000G 2$: MOV #BTOVER,R2 ;POINT TO BT OVERRIDE TABLE
451 001132 016703 000000G MOV BINWD,R3 ;LOAD WORD OFFSET
452 001136 006303 ASL R3 ;SHIFT FOR BYTE OFFSET
453 001140 060302 ADD R3,R2 ;START LIST HERE
454 ;
455 ;
456 ;
457 ;
458 ;
459 001142 012705 000000G 3$: MOV #PRINT,R5 ;POINT TO PRINT LINE
460 001146 012703 000005 MOV #5,R3 ;COUNT NUMBER OF WORDS IN PRINT LINE
461 001152 012201 4$: MOV (R2)+,R1 ;LOAD WORD FROM BT OVERRIDE TABLE
462 001154 004767 000000G JSR PC,UNPK ;CONVERT FOR PRINTING
463 001160 020227 000000G CMP R2,#BTOVER+(BTEND-BTRANS) ;END OF TABLE REACHED
464 001164 103011 BHIS 5$ ;YES, EXIT
465 001166 005303 DEC R3 ;SUB FROM PRINT LINE COUNT
466 001170 001370 BNE 4$ ;CONVERT NEXT TABLE ENTRY
467 001172 004767 000000G JSR PC,CONSOL ;PRINT LINE
468 001176 032767 000000G-000000G BIT #LOOP,BASE ;LOOP ON
469 001204 001403 BEQ LI1X ;NO, EXIT
470 001206 000755 BR 3$ ;START WITH FRESH LINE
471 ;
472 001210 004767 000000G 5$: JSR PC,CONSOL ;PRINT LAST LINE
473 ;
474 ;
475 001214 004767 000000G LI1X: JSR PC,KILL ;KILL AST (IF THERE WAS ONE)
476 001220 000167 000000G JMP COMXX

```

```

478      ;
479      ;
480      ;
481      ;
482      ;
483      ;
484      ;
485      ;
486      ;
487      ;
488      ;
489      ;
490      ;
491      ;
492      ;
493      ;
494      001224      ;
495      001224      004767      000000G      EN1::      JSR      PC,FIND      ;LOCATE END VALUE IN COMMAND LINE
496      001230      103003      ;
497      001232      004767      000000G      BCC      3$      ;OK, CONTINUE
498      001236      000425      ;
499      001240      004767      000000G      BR      EN1X      ;MISSING OPERAND
500      001244      103003      3$:      JSR      PC,PACK      ;EXIT
501      001246      004767      000000G      BCC      4$      ;CONVERT COMMAND LINE VALUE TO BINARY
502      001252      000417      ;
503      ;
504      ;
505      ;
506      001254      005767      000000G      JSR      PC,ERR4      ;CONVERSION SUCCESSFUL
507      001260      001003      ;
508      001262      004767      000000G      BR      EN1X      ;INVALID NUMERIC VALUE
509      001266      000411      ;
510      ;
511      ;
512      ;
513      ;
514      ;
515      ;
516      001270      005767      000000G      THE END VALUE CANNOT BE ZERO.
517      001274      001003      4$:      TST      BINWD      ;IS END VALUE ZERO
518      001276      004767      000000G      BNE      5$      ;OK, CONTINUE
519      001302      000403      ;
520      001304      016767      000000G 000000G 6$:      JSR      PC,ERR4      ;REPORT INVALID VALUE
521      ;
522      001312      ;
523      001312      000167      000000G      BR      EN1X      ;AND EXIT

```

THE 'DATA' COMMAND MUST HAVE BEEN PREVIOUSLY EXECUTED.
IF IT HAS BEEN, REPLACE THE CONTENTS OF THE FIELD
'DATALN' (LENGTH OF LOCAL DATA BUFFER) WITH THE
END VALUE FROM THE COMMAND LINE.

```

516      001270      005767      000000G      5$:      TST      DATALN      ;HAS 'DATA' COMMAND BEEN EXECUTED
517      001274      001003      ;
518      001276      004767      000000G      BNE      6$      ;YES, CONTINUE
519      001302      000403      ;
520      001304      016767      000000G 000000G 6$:      JSR      PC,ERR10      ;DATA COMMAND MUST PRECEED 'END'
521      ;
522      001312      ;
523      001312      000167      000000G      BR      EN1X      ;REPLACE LENGTH OF 'DATA'

```

JMP COMXX


```

525      ;
526      ;
527      ; DATA FILES ROUTINES.
528      ;
529      ;
530      ; READ EITHER THE 'DATA' BUFFER OR THE BYTE TRANSLATOR
531      ; BUFFER FROM DISK.
532      ; PERFORM SECOND LEVEL PARSING.
533      ; EG. IN THE COMMAND:
534      ; >DF IB
535      ; PARSE THE 'IB'
536      ;
537      ; DF1::
538      001316      004767      000000G      JSR      PC,FIND      ;FIND A BUFFER MNEMONIC IN COMMAND LINE
539      001322      103004      000000G      BCC      1$          ;OK, CONTINUE
540      001324      004767      000000G      JSR      PC,ERR3      ;'MISSING OPERAND'
541      001330      000167      000514      JMP      DFIX
542      ;
543      ; MATCH THE MNEMONIC FROM THE COMMAND LINE AGAINST THE
544      ; TABLE OF VALID MNEMONICS.
545      ;
546      001334      012700      000002      1$: MOV      #DF1LN,R0      ;NUMBER OF TABLE ENTRIES
547      001340      012702      000020*      MOV      #DF1TBL,R2      ;TABLE OF RESPONSES/RTH ADDRESSES
548      001344      004767      000000G      JSR      PC,SCAN      ;CHECK COMMAND LINE AGAINST TABLE
549      001350      103004      000000G      BCC      2$          ;OK, CONTINUE
550      001352      004767      000000G      JSR      PC,ERR11      ;'INVALID FILE MNEMONIC'
551      001356      000167      000466      JMP      DFIX
552      001362      000171      000000G      2$: JMP      @R1          ;JUMP TO ROUTINE
553      ;
554      ;
555      ; LOAD 'DATA' BUFFER FROM DISK.
556      ; THIS ROUTINE READS A 'DATA' FILE FROM DISK ONE RECORD AT A
557      ; TIME. IT THEN CALLS A SUBROUTINE IN TMAIN WHICH PROCESSES
558      ; THE LINE AND PLACES IT INTO TWO BUFFERS. ONE OF THE BUFFERS
559      ; IS SIMPLY A MIRROR OF THE INPUT LINE INPUT TO BE USED FOR
560      ; BUFFER VERIFICATION. THE OTHER BUFFER IS THE 6-BIT BYTE
561      ; BUFFER DETECTOR WILL EVENTUALLY BE TRANSFERRED TO THE
562      ; TERM DETECTOR INPUT BUFFER (SEE THE 'TR' AND 'RU'
563      ; COMMANDS).
564      ;
565      ; CLEAR LOCAL DATA BUFFER (BOTH THE 6-BIT BYTE TABLE
566      ; AND THE 8-BIT ASCII MIRROR TABLE).
567      ;
568      001366      005067      000000G      DF1B:: CLR      COUNT      ;CLEAR BIT SHIFT COUNTER
569      001372      005067      000000G      CLR      DATALN      ;CLEAR LENGTH OF 'DATA'
570      001376      012700      000000G      MOV      #DTBL,R0      ;POINT TO 6-BIT BYTE TABLE
571      001402      012701      001000      MOV      *(<<1364.*3>/8.)+1,R1 ;NUMBER OF WORDS IN TABLE
572      001406      005020      000000G      1$: CLR      (R0)+      ;CLEAR DATA TABLE
573      001410      005301      000000G      DEC      R1
574      001412      001375      000000G      BNE      1$
575      ;
576      ;
577      001414      012700      000000G      2$: MOV      #DSAVE,R0      ;POINT TO ASCII MIRROR TABLE
578      001420      012701      001252      MOV      #1364/2,R1      ;NUMBER OF WORDS
579      001424      005020      000000G      CLR      (R0)+      ;CLEAR TABLE
580      001426      005301      000000G      DEC      R1
581      001430      003375      000000G      BGT      2$

```

```

582.      ;
583      ;
584      ;
585 001432 012704 000000G.      MOV.      #DTBL,R4      ;R4 -> 6-BIT-BYTE-TABLE.
586 001436 012705 000000G.      MOV.      #DSAVE,R5      ;R5 -> 8-BIT-ASCII-MIRROR-TABLE.
587 001442      ;
      001442 012700 000000G.      OPEN$R.      #TXTFDB.
      001446 112760 000000G-000000G.      MOV.      #TXTFDB,R0
      001454 004767 000000G.      .IIF.      NB,#FO,RD,      MOV$ #FO,RD,F.FACC(R0)
588 001460 103005      JSR.      PC,.OPEN.
589 001462 004767 000000G.      BCC.      4$
590 001466 005067 000000G.      JSR.      PC,ERR12.      ;ERROR-ON-OPEN.
591 001472 000566      CLR.      DATALN.      ;AS-THOUGH-RTN-WAS-NOT-ENTERED.
592      BR.      DF1X
593      ;
594      ;
595      ;
596 001474      4$:      GET$      #TXTFDB.      ;READ-'DATA'-FILE.
      001474 012700 000000G.      MOV.      #TXTFDB,R0
      001500 004767 000000G.      JSR.      PC,.GET.
597 001504 103414      BCS.      5$      ;ERROR-OR-END-OF-FILE.
598 001506 016067 000000G-000000G.      MOV.      F,NRBD(R0),GCMLN. ;LENGTH-OF-LINE-READ.
599 001514 012701 000000G.      MOV.      #GCM$BUF,R1      ;POINT-TO-LINE.
600 001520 004767 000000G.      JSR.      PC,DBLD.      ;BUILD-6-BIT-BYTE-TABLE.
601 001524 103551      BCS.      DF1X      ;ERROR-OR-END-OF-TABLE.
602 001526 006767 000000G-000000G.      ADD.      DPLUS,DATALN.      ;ACCUMULATE-LENGTH.
603 001534 000757      BR.      4$      ;GET-NEXT-LINE.
604      ;
605 001536 122760 177766 000000G-5$:      CMPB.      #-10,,F.ERR(R0) ;END-OF-FILE.
606 001544 001405      BEQ.      6$      ;YES,PUT-OUT-MESSAGE.
607 001546 004767 000000G.      JSR.      PC,ERR13      ;ERROR-ON-GET.
608 001552 005067 000000G.      CLR.      DATALN.      ;AS-THOUGH-RTN-WAS-NOT-ENTERED.
609 001556 000402      BR.      7$      ;CLOSE-FILE-AND-EXIT.
610      ;
611 001560 004767 000000G.      6$:      JSR.      PC,ENFILE.
612 001564      7$:      CLOSE$      #TXTFDB.
      001564 012700 000000G.      MOV.      #TXTFDB,R0
      001570 004767 000000G.      JSR.      PC,.CLOSE.
613 001574 000525      BR.      DF1X
614      ;
615      ;
616      ;
617      ;
618      ;
619      ;
620      ;
621      ;
622 001576      DFBT:
623 001576 012700 000000G.      MOV.      #BTOVER,R0      ;POINT-TO-BYTE-TRANS-TABLE.
624 001602 012701 000100      MOV.      #64,,R1      ;NUMBER-OF-WORDS.
625 001606 005020      1$:      CLR.      (R0)+      ;CLEAR-TABLE.
626 001610 005301      DEC.      R1
627 001612 003375      BGT.      1$
628      ;
629      ;
630      ;
631 001614      OPEN.      'TRANS.TXT'
      OPEN$R.      #TRFDB.

```

```
001614 012700 000000G MOV. #TRFDB,R0
001620 112760 000000G 000000G .IIF. NB,*FO,RD, MOVB,*FO,RD,F,FACC(R0)
001626 004767 000000G JSR. PC,.OPEN
632 001632 103003 BCC. 2$
633 001634 004767 000000G JSR. PC,ERR14 ;ERROR ON OPEN
634 001640 000503 BR. DF1X
635 ;
636 ;
637 ;
638 001642 012702 000000G 2$: MOV. #BTOVER,R2 ;R2-> BYTE TRANS TABLE
639 001646 000000G 3$: GET$ #TRFDB
001646 012700 000000G MOV. #TRFDB,R0
001652 004767 000000G JSR. PC,.GET
640 001656 103021 BCC. 6$ ;OK, PROCESS RECORD
641 001660 122760 177766 000000G CMPB. #-10,,F,ERR(R0) ;END OF FILE
642 001666 001403 BEQ. 20$ ;YES, EXIT
643 001670 004767 000000G JSR. PC,ERR15 ;ERROR ON GET
644 001674 000447 BR. 30$ ;CLOSE FILE
645 ;
646 ;
647 ;
648 001676 052767 000000G 000000G 20$: BIS. #RIDE,BASE ;SET FLAG FOR OVERRIDE TABLE FILLED
649 001704 004767 000000G JSR. PC,ENFILE ;PRINT END-OF-FILE MESSAGE
650 001710 CLOSE$ #TRFDB
001710 012700 000000G MOV. #TRFDB,R0
001714 004767 000000G JSR. PC,CLOSE
651 001720 000453 BR. DF1X ;EXIT
652 ;
653 ;
654 ;
655 ;
656 ;
657 ;
658 001722 016067 000000G 000000G 6$: MOV. F,NRBD(R0),GCMLN ;PRETEND DISK RECORD IS A COMMAND LINE
659 001730 012701 000000G MOV. #GCMBUF,R1 ;POINT TO RECORD READ
660 001734 004767 000000G JSR. PC,FIND ;LOCATE A TABLE VALUE
661 001740 103003 BCC. 7$ ;OK, CONTINUE
662 001742 004767 000000G JSR. PC,ERR16 ;EMPTY INPUT LINE
663 001746 000422 BR. 30$ ;CLOSE FILE AND EXIT
664 ;
665 001750 004767 000000G 7$: JSR. PC,PACK ;CONVERT NUMERIC VALUE FROM DISK RECORD
666 001754 103003 BCC. 8$ ;OK, CONTINUE
667 001756 004767 000000G JSR. PC,ERR4 ;INVALID NUMERIC VALUE
668 001762 000414 BR. 30$ ;CLOSE FILE AND EXIT
669 ;
670 001764 020227 000000G 8$: CMP. R2,#BTOVER+<BTEND-BTRANS> ;END OF TABLE REACHED
671 001770 103403 BLO. 10$ ;NO, CONTINUE
672 001772 004767 000000G JSR. PC,ERR17 ;TABLE OVERFLOW
673 001776 000406 BR. 30$ ;CLOSE FILE AND EXIT
674 002000 016722 000000G 10$: MOV. BINWD,(R2)+ ;ENTER WORD IN BT TABLE
675 002004 004767 000000G JSR. PC,FIND ;LOCATE NEXT TABLE VALUE
676 002010 103357 BCC. 7$ ;PROCESS STRING
677 002012 000715 BR. 3$ ;NOTHING THERE, FINISHED WITH THIS RECORD
678 ;
679 ;
680 ;
681 002014 30$: CLOSE$ #TRFDB
```

TDATA- MACRO-M1110 27-MAR-80 15:36 PAGE 9-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
      002014 012700 000000G-      MOV-  #TRFDB,R0
      002020 004767 000000G-      JSR-  PC,,CLOSE-
682-      ;
683 002024 042767 000000G-000000G-      BIC-  #RIDE,BASE-      ;CLEAR-BT-OVERRIDE FLAG-
684 002032 012700 000000G-      MOV-  #BTOVER,R0      ;POINT TO-OVERRIDE TABLE-
685 002036 012701 000400      MOV-  #256,,R1      ;NUMBER OF WORDS IN TABLE-
686 002042 005020      35$: CLR-  (R0)+      ;CLEAR TABLE-
687 002044 005301      DEC-  R1
688 002046 001375      BNE-  35$
689      ;
690      ;
691 002050      DF1X:
692 002050 000167 000000G-      JMP-  COMXX-
```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

694      ;
695      ;
696      ;      TRANSFER
697      ;      MOVE LOCAL DATA TABLE TO TERM DETECTOR INPUT BUFFER
698      ;      OR BYTE TRANSLATOR OVERRIDE TABLE TO TERM DETECTOR
699      ;      BYTE TRANSLATOR
700      ;      PERFORM SECOND LEVEL PARSING
701      ;      EG. IN THE COMMAND
702      ;      >TR IB
703      ;      PARSE THE 'IB'
704      ;
705      ;
706      002054      TR1::
707      002054      004767      000000G      JSR      PC,FIND      ;FIND THE BUFFER MNEMONIC IN COMMAND LINE
708      002060      103003      BCC      1$      ;OK, CONTINUE
709      002062      004767      000000G      JSR      PC,ERR3      ;'MISSING OPERAND'
710      002066      000532      BR      TR1X
711      ;
712      ;      MATCH THE MNEMONIC IN THE COMMAND LINE AGAINST THE
713      ;      TABLE OF VALID MNEMONICS
714      ;
715      002070      012700      000002      1$:      MOV      #TR1LN,R0      ;NUMBER OF TABLE ENTRIES
716      002074      012702      000030      MOV      #TR1BL,R2      ;TABLE OF RESPONSES/RTH ADDRESSES
717      002100      004767      000000G      JSR      PC,SCAN      ;MATCH COMMAND LINE AGAINST TABLE
718      002104      103003      BCC      2$      ;OK, CONTINUE
719      002106      004767      000000G      JSR      PC,ERR10     ;'INVALID TABLE NAME'
720      002112      000520      BR      TR1X
721      ;
722      ;      SAVE THE POINTER TO THE ROUTINE ASSOCIATED WITH THE
723      ;      BUFFER. R1 -> ROUTINE ADDRESS
724      ;      CALL A ROUTINE TO SCAN THE COMMAND LINE FOR A LOOP
725      ;      INDICATOR. EG:
726      ;      >TR IB-L
727      ;      LOOP FLAG WILL BE SET IF INDICATOR IS PRESENT
728      ;
729      002114      010167      000000G      2$:      MOV      R1,R1NPT      ;SAVE POINTER TO ROUTINE
730      002120      004767      000000G      JSR      PC,LOOPR      ;LOOP ON TRANSFER
731      002124      016701      000000G      TRIN:      MOV      R1NPT,R1      ;POINT TO RTH
732      002130      000171      000000      JMP      @R1      ;JUMP TO ROUTINE
733      ;
734      ;
735      ;      TRANSFER FROM LOCAL DATA BUFFER TO TERM DETECTOR
736      ;      INPUT BUFFER. THE FIELD 'DATALN' CONTAINS THE NUMBER
737      ;      OF 6-BIT BYTE CHARACTERS IN THE LOCAL BUFFER. CONVERT
738      ;      THIS VALUE INTO A NUMBER PDP-11 WORDS FOR CONTROLLING
739      ;      THE TRANSFER.
740      ;
741      ;
742      002134      TRIB::
743      002134      012767      000000      176376      MOV      #0,TD$SW      ;RESET TD
744      002142      012767      000200      176360      MOV      #TD$DISK,TD$CTW      ;SIMULATE DISK INPUT
745      002150      016701      000000G      MOV      DATALN,R1      ;NUMBER OF 6-BIT BYTES
746      002154      006301      MOV      R1      ;MULT BY 2
747      002156      016702      000000G      MOV      DATALN,R2      ;NUMBER OF 6-BIT BYTES
748      002162      006201      ADD      R2,R1      ;RESULT = NUMBER OF BYTES X 3
749      002164      006201      ASR      R1      ;NOW DIVIDE BY 8
750      002166      006201      ASR      R1      ;FOR NUMBER

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

751 002170 006201 ASR R1 ;OF WORDS
752 002172 016767 000000G 000000G MOV ILOW,MSTR2 ;IB START ADDRESS
753 002200 012702 000000G MOV #DTBL,R2 ;R2 -> 6-BIT BYTE TABLE
754 ;
755 ;
756 ; TRANSFER THE TABLE
757 002204 016703 000000G 1$: MOV MSTR2,R3 ;POINT TO IB
758 002210 012213 MOV (R2)+,(R3) ;LOAD IB
759 002212 062767 000002 000000G ADD #2,MSTR2 ;ADVANCE ADDRESS
760 002220 005301 DEC R1
761 002222 003370 BGT 1$
762 002224 000447 BR TRLP ;TEST LOOP FLAG
763 ;
764 ;
765 ; TRANSFER BYTE TRANSLATOR TABLE TO TD
766 ;
767 ;
768 ;
769 ; IF A BYTE TRANSLATOR OVERRIDE TABLE HAS BEEN BUILT
770 ; (SEE THE 'CH' AND 'DF' COMMANDS), THEN TRANSFER IT
771 ; TO THE TERM DETECTOR, OTHERWISE TRANSFER THE DEFAULT
772 ; BYTE TRANSLATOR TABLE. IN EITHER CASE TRANSFER THE
773 ; SAME TABLE FOUR TIMES.
774 002226 TRBT::
775 002226 012767 000000 176376 MOV #0,TD$SW ;RESET TD
776 002234 012702 000004 MOV #4,R2 ;TRANSFER SAME TABLE 4 TIMES
777 002240 016767 000000G 000000G MOV TLOW,MSTR2 ;LOAD START ADDRESS
778 002246 012700 000100 1$: MOV #64,R0 ;NUMBER OF WORDS IN TABLE
779 002252 032767 000000G 000000G BIT #RIDE,BASE ;IS OVERRIDE TABLE IN USE
780 002260 001403 BEQ 2$ ;NO
781 002262 012701 000000G MOV #BTOVER,R1 ;POINT TO OVERRIDE TABLE
782 002266 000402 BR 3$
783 002270 012701 000000G 2$: MOV #BTRANS,R1 ;R1 -> BYTE TRANS TABLE
784 002274 012767 000040 176360 3$: MOV #T$ICD,TD$CTW ;SET INTERFACE AND CONTROL
785 002302 012767 000030 176362 4$: MOV #T$BTAR,TD$TAW ;SELECT BYTE TRANS MAR
786 002310 016767 000000G 176364 MOV MSTR2,TD$TDW
787 002316 012767 000020 176362 MOV #T$BT,TD$TAW ;SELECT BYTE TRANS MEMORY
788 002324 012167 176364 MOV (R1)+,TD$TDW
789 002330 005267 000000G INC MSTR2 ;ADVANCE ADDRESS
790 002334 005300 DEC R0
791 002336 003361 BGT 4$
792 002340 005302 DEC R2
793 002342 001341 BNE 1$ ;SUB FROM OUTER LOOP COUNT
794 ;
795 002344 TRLP:
796 002344 032767 000000G 000000G BIT #LOOP,BASE ;LOOP ON TRANSFER
797 002352 001264 BNE TRIN ;YES, REPEAT
798 ;
799 002354 TR1X:
800 002354 004767 000000G JSR PC,KILL ;KILL AN AST (IF THERE WAS ONE)
801 002350 000167 000000G JMP COMXX
802 ;
803 000001 .END

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
ALUCKE = 040000
ALUOE = 004000
A01 = 010000
BASE = ***** GX
BINWD = ***** GX
BITVAL = 000000
BIT0 = 000001
BIT1 = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2 = 000004
BIT3 = 000010
BIT4 = 000020
BIT5 = 000040
BIT6 = 000100
BIT7 = 000200
BIT8 = 000400
BIT9 = 001000
BTEND = ***** GX
BTOVER = ***** GX
BTRANS = ***** GX
BYTE0 = 000000
BYTE1 = 000001
BYTE10 = 000012
BYTE11 = 000013
BYTE12 = 000014
BYTE13 = 000015
BYTE14 = 000016
BYTE15 = 000017
BYTE16 = 000020
BYTE17 = 000021
BYTE18 = 000022
BYTE19 = 000023
BYTE2 = 000002
BYTE20 = 000024
BYTE21 = 000025
BYTE22 = 000026
BYTE23 = 000027
BYTE24 = 000030
BYTE25 = 000031
BYTE26 = 000032
BYTE27 = 000033
BYTE28 = 000034
BYTE29 = 000035
BYTE3 = 000003
BYTE30 = 000036
BYTE31 = 000037
BYTE32 = 000040
BYTE33 = 000041
BYTE34 = 000042
BYTE35 = 000043
BYTE36 = 000044
BYTE37 = 000045
BYTE38 = 000046
BYTE39 = 000047
BYTE4 = 000004
BYTE40 = 000050
BYTE41 = 000051
BYTE42 = 000052
BYTE43 = 000053
BYTE44 = 000054
BYTE45 = 000055
BYTE46 = 000056
BYTE47 = 000057
BYTE48 = 000060
BYTE49 = 000061
BYTE5 = 000005
BYTE50 = 000062
BYTE51 = 000063
BYTE52 = 000064
BYTE53 = 000065
BYTE54 = 000066
BYTE55 = 000067
BYTE56 = 000070
BYTE57 = 000071
BYTE58 = 000072
BYTE59 = 000073
BYTE6 = 000006
BYTE60 = 000074
BYTE61 = 000075
BYTE62 = 000076
BYTE63 = 000077
BYTE64 = 000100
BYTE65 = 000101
BYTE66 = 000102
BYTE67 = 000103
BYTE68 = 000104
BYTE69 = 000105
BYTE7 = 000007
BYTE70 = 000106
BYTE71 = 000107
BYTE72 = 000110
BYTE73 = 000111
BYTE74 = 000112
BYTE75 = 000113
BYTE76 = 000114
BYTE77 = 000115
BYTE78 = 000116
BYTE79 = 000117
BYTE8 = 000008
BYTE80 = 000120
BYTE81 = 000121
BYTE82 = 000122
BYTE83 = 000123
BYTE84 = 000124
BYTE85 = 000125
BYTE86 = 000126
BYTE87 = 000127
BYTE88 = 000130
BYTE89 = 000131
BYTE9 = 000011
BYTE90 = 000132
BYTE91 = 000133
BYTE92 = 000134
BYTE93 = 000135
BYTE94 = 000136
BYTE95 = 000137
BYTE96 = 000140
BYTE97 = 000141
BYTE98 = 000142
BYTE99 = 000143
BYTVAL = 000144
CBKALL = 001000
CBKCLK = 000400
CHBT = 000376RG
CHIB = 000260RG
CHLEN = ***** GX
CH1 = 000150RG
CH1LN = 000002 G
CH1TBL = 000000RG
CH1X = 000546R
CNOBRE = 100000
COMXX = ***** GX
CONSOL = ***** GX
COUNT = ***** GX
CPCCEN = 010000
CPREAD = 040000
CPWRITE = 020000
CSABRD = 000004
CSEGC1 = 100000
CSOE = 000040
CSWRTE = 000100
DATA = 000114R
DATALN = ***** GX
DA1 = 000040RG
DA1X = 000144R
DBLD = ***** GX
DBR RD = 000001
DBCPP = 001457
DBSPPT = 000026
DBSTPC = 000023
DFBT = 001576RG
DFIB = 001366RG
DF1 = 001316RG
DF1LN = 000002 G
DF1TBL = 000020RG
DF1X = 002050R
DISPCS = 100000
DMAWR = 000005
DMARRD = 000003
DMARWR = 000004
DPLUS = ***** GX
DSAVE = ***** GX
DTBL = ***** GX
ENBR = 010000
ENFILE = ***** GX
EN! = 001224RG
EN1X = 001312R
ERR10 = ***** GX
ERR11 = ***** GX
ERR12 = ***** GX
ERR13 = ***** GX
ERR14 = ***** GX
ERR15 = ***** GX
ERR16 = ***** GX
ERR17 = ***** GX
ERR18 = ***** GX
ERR20 = ***** GX
ERR22 = ***** GX
ERR3 = ***** GX
ERR4 = ***** GX
ERR6 = ***** GX
ERR7 = ***** GX
ERR8 = ***** GX
ERR9 = ***** GX
FIND = ***** GX
FORD = ***** GX
FERR = ***** GX
FACC = ***** GX
FNRBD = ***** GX
GCMBUF = ***** GX
GCMLEN = ***** GX
GCONLY = ***** GX
HANG = ***** GX
ILOW = ***** GX
KILL = ***** GX
LIBT = 001072RG
LIIB = 000672RG
LI1 = 000552RG
002 LI1LN = 000002 G
LI1TBL = 000010RG
002 LI1X = 001214R
002 LOC EN = 000100
LOC WA = 040000
LOC WB = 100000
LOOP = ***** GX
LOOPR = ***** GX
MAREN1 = 000001
002 MAREN2 = 004000
002 MARLOD = 010000
002 MAROUT = 000002
MAR LO = 002000
002 MAR OU = 000040
002 MBKALL = 001000
MBKCLK = 000400
MMABRD = 000100
MMLEFT = 000002
MMOE = 000004
MMWRTE = 000010
MNOBRE = 100000
MREN1 = 000001
MREN2 = 020000
MSTR2 = ***** GX
002 MSYN = 000040
002 N = 000144
OFFSET = ***** GX
OFF6 = ***** GX
PACK = ***** GX
PAR$$$ = 000027
PLB = 000010
PLC = 000020
PLD = 000030
PLRWR = 000200
PLREN = 000200
PRINT = ***** GX
QR$CR1 = 176420
QR$CR2 = 176422
QR$LBR = 176424
Q$ATTN = 000100
Q$BCL = 000001
Q$CCCP = 000040
Q$CHB = 000400
Q$CHRL = 000200
Q$CLR = 000040
Q$CNC = 030000
Q$CP = 000000
Q$CPCC = 000010
Q$CP2 = 000260
Q$CSC = 010000
Q$CSEL = 000360
Q$CSET = 000002
Q$CSP = 020000
Q$DMA = 000001
002 Q$ENBK = 040000
002 Q$ENOP = 020000
002 Q$FAL = 004000
002 Q$FC = 000045
002 Q$FO = 000044
002 Q$FP = 000047
Q$HBF = 000002
Q$ICP = 000006
Q$IHB = 000003
Q$IHRL = 000002
Q$IMRP = 000007
Q$LBD = 001000
Q$LBDP = 001001
Q$LBP = 000001
Q$LCD = 000003
Q$LMD = 000004
Q$LDPP = 002000
Q$LHP = 010000
Q$MNC = 140000
Q$MR = 000052
Q$MRP = 000040
Q$MRP2 = 000240
Q$MSC = 040000
Q$MSET = 000004
Q$MSP = 100000
Q$NCLK = 176000
Q$PP = 000100
Q$PPSW = 000320
```

TDATA: MACRO-M1110 27-MAR-80 15:36 PAGE 10-3
SYMBOL TABLE

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
Q$PP2 = 000300      TD$TDW = 176364      T$RBD0 = 000200      WORD32 = 000100      WORD69 = 000212
Q$QHLT = 000013      TLOW = ***** GX.  T$RNB = 000040      WORD33 = 000102      WORD7 = 000016
Q$QL = 000043        TRBT = 002226RG      002 T$RSET = 040000      WORD34 = 000104      WORD70 = 000214
Q$QLA = 000053       TRFDB = ***** GX.  T$SC = 000022      WORD35 = 000106      WORD71 = 000216
Q$QLB = 000054       TRIB = 002134RG      002 T$SCLK = 020000      WORD36 = 000110      WORD72 = 000220
Q$QLR = 000001       TRIN = 002124R      002 T$SEG1 = 000000      WORD37 = 000112      WORD73 = 000222
Q$QW = 000042        TRLP = 002344R      002 T$SEG2 = 000001      WORD38 = 000114      WORD74 = 000224
Q$RDCD = 000005      TR1 = 002054RG      002 T$SEG3 = 000002      WORD39 = 000116      WORD75 = 000226
Q$RDMD = 000006      TR1LN = 000002 G    T$SO = 000001      WORD4 = 000010      WORD76 = 000230
Q$REBK = 001000      TR1TBL 000030RG      002 T$UBUS = 100000      WORD40 = 000120      WORD77 = 000232
Q$RNC = 006000       TR1X = 002354R      002 T$1CLK = 000400      WORD41 = 000122      WORD78 = 000234
Q$RSC = 004000       TXTFDB = ***** GX.  T$BBEN = 000020      WORD42 = 000124      WORD79 = 000236
Q$RSET = 000010      T$AD = 000020      UBD.IN = 000020      WORD43 = 000126      WORD8 = 000020
Q$SM = 100000        T$BA = 000002      UNPK = ***** GX.  WORD44 = 000130      WORD80 = 000240
Q$SP = 000120        T$BD = 000010      WORD0 = 000000      WORD45 = 000132      WORD81 = 000242
Q$SP2 = 000340       T$BS0 = 100000      WORD1 = 000002      WORD46 = 000134      WORD82 = 000244
RGQ.EN = 000200      T$BT = 000020      WORD10 = 000024      WORD47 = 000136      WORD83 = 000246
RGQ.VA = 020000      T$BTAR = 000030      WORD11 = 000026      WORD48 = 000140      WORD84 = 000250
RIDE = ***** GX.  T$BTD = 002000      WORD12 = 000030      WORD49 = 000142      WORD85 = 000252
RTNPT = ***** GX.  T$CD = 000100      WORD13 = 000032      WORD5 = 000012      WORD86 = 000254
SCAN = ***** GX.  T$CLK = 002000      WORD14 = 000034      WORD50 = 000144      WORD87 = 000256
SEQ.CI = 000010      T$DISK = 000200      WORD15 = 000036      WORD51 = 000146      WORD88 = 000260
S$CLR = 000000      T$DRD = 000004      WORD16 = 000040      WORD52 = 000150      WORD89 = 000262
S$LA = 000001      T$MEM = 010000      WORD17 = 000042      WORD53 = 000152      WORD9 = 000022
S$OB = 000005      T$FSAA = 000000      WORD18 = 000044      WORD54 = 000154      WORD90 = 000264
S$OR = 000006      T$FSAB = 000004      WORD19 = 000046      WORD55 = 000156      WORD91 = 000266
S$QX = 000004      T$FSAC = 000014      WORD2 = 000004      WORD56 = 000160      WORD92 = 000270
S$SR = 000007      T$FSB2 = 000010      WORD20 = 000050      WORD57 = 000162      WORD93 = 000272
S$S1 = 000010      T$IB = 000026      WORD21 = 000052      WORD58 = 000164      WORD94 = 000274
S$S2 = 000014      T$IBAR = 000024      WORD22 = 000054      WORD59 = 000166      WORD95 = 000276
TD$CTR = 176370      T$IBE = 020000      WORD23 = 000056      WORD6 = 000014      WORD96 = 000300
TD$CTW = 176360      T$IBF = 040000      WORD24 = 000060      WORD60 = 000170      WORD97 = 000302
TD$INL = 004000      T$ICD = 000040      WORD25 = 000062      WORD61 = 000172      WORD98 = 000304
TD$MEM = 000270      T$MODE = 004000      WORD26 = 000064      WORD62 = 000174      WORD99 = 000306
TD$OAR = 176344      T$OB = 000036      WORD27 = 000066      WORD63 = 000176      WRDVAL = 000310
TD$OTR = 176346      T$OBE = 004000      WORD28 = 000070      WORD64 = 000200      XTREAD = 001000
TD$ORD = 000274      T$OBF = 010000      WORD29 = 000072      WORD65 = 000202      XTWRITE = 000400
TD$SW = 176376      T$OBRA = 000034      WORD3 = 000006      WORD66 = 000204      .CLOSE = ***** G.
TD$TAR = 176372      T$OBWA = 000032      WORD30 = 000074      WORD67 = 000206      .GET = ***** G.
TD$TAW = 176362      T$OUTH = 100000      WORD31 = 000076      WORD68 = 000210      .OPEN = ***** G.
TD$TDR = 176374
```

```
. ABS. 000000 000
000000 001
TDATA 002364 002
ERRORS DETECTED: 0
```

VIRTUAL MEMORY USED: 4595 WORDS (18 PAGES)
DYNAMIC MEMORY: 5972 WORDS (22 PAGES)
ELAPSED TIME: 00:01:01
IDATA,TDATA,SP=C20,1JIM,C20,1JTDATA

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3


```
1 .TITLE .TRUN
2 000000 .PSECT .TRUN
3 .LIST .MEB
4
5
6
7
8
9
10
11
12
13
14
15
16
17 000000
18 000000 123 103
19 000002 000070 120
20 000004 121 120
21 000006 000124 116
22 000010 121 116
23 000012 000134 103
24 000014 116 103
25 000016 000144 103
26 000004
27
28
29
30
31
32
33
34
35 000020
36 000020 004767 000000
37 000024 103004
38 000026 004767 000000
39 000032 000167 000532
40
41
42
43
44 000036 012700 000004
45 000042 012702 000000
46 000046 004767 000000
47 000052 103004
48 000054 004767 000000
49 000060 000167 000504
50
51
52
53 000064 000171 000000
54
55
56
57
```

```
.TERM DETECTOR 'MANUAL' DEBUGGING AIDS
.RUN COMMANDS

COMMANDS:
RU .RUN TERM DETECTOR

ONCE A COMMAND HAS BEEN EXECUTED (OR AN ERROR ENCOUNTERED)
THIS MODULE RETURNS CONTROL TO THE MODULE TMAIN AT LOCATION
'COMXX'.

RU1TBL::
.ASCII /SC/ ;SINGLE CLOCK
.WORD RUSC
.ASCII /QP/ ;READ OUTPUT TO QR AND PRINT
.WORD RUQP
.ASCII /QN/ ;READ OUTPUT TO QR (NO PRINT)
.WORD RUQN
.ASCII /NC/ ;NORMAL CONTINUOUS CLOCK
.WORD RUNC
RU1LN. == <.-RU1TBL>/4

RUN
PERFORM SECOND LEVEL PARSING
EG. IN THE COMMAND:
>RU NC
PARSE THE 'NC'

RU1::
JSR PC,FIND ;FIND AN OPERAND IN COMMAND LINE
BCC 1$ ;OK, CONTINUE
JSR PC,ERR3 ;MISSING OPERAND
JMP RU1X

MATCH THE OPERAND FROM THE COMMAND LINE AGAINST A TABLE
OF VALID OPERANDS:

1$: MOV #RU1LN,R0 ;NUMBER OF TABLE ENTRIES
MOV #RU1TBL,R2 ;TABLE OF RESPONSES/RTN ADDRESSES
JSR PC,SCAN ;MATCH COMMAND LINE AGAINST TABLE
BCC 2$ ;OK, CONTINUE
JSR PC,ERR18 ;INVALID TABLE NAME
JMP RU1X

R1 -> ADDRESS OF ROUTINE TO BE EXECUTED.

2$: JMP @R1 ;JUMP TO ROUTINE

RUN WITH A SINGLE CLOCK
```

```
58      ; CALL A ROUTINE TO SCAN THE COMMAND LINE FOR A
59      ; LOOP INDICATOR. EG:
60      ; >RU SC L.
61      ; LOOP FLAG WILL BE SET IF INDICATOR IS PRESENT.
62      ;
63      ;
64      000070      RUSC::
65      000070      004767 000000G. JSR PC,LOOPR. ;LOOP ON CLOCK ?
66      000074      012767 000000 176376 MOV #0,TD$SW. ;RESET TD
67      000102      012767 024000 176360 1$: MOV #(<T$CLK+T$MODE>),TD$CTW.
68      000110      032767 000000G.000000G. BIT #LOOP,BASE. ;LOOP FLAG ON ?
69      000116      001371 BNE 1$ ;YES, REPEAT.
70      000126      000167 000444 JMP RUIX
71      ;
72      ;
73      ; RUN NORMAL.
74      ; READ TD OUTPUT TO QR AND PRINT.
75      ;
76      ;
77      000124      RUQP::
78      000124      052767 000000G.000000G. BIS #QP,BASE. ;SET FLAG FOR PRINT
79      000132      000404 BR RUNC ;RUN NORMAL.
80      ;
81      ;
82      ; RUN NORMAL.
83      ; READ TD OUTPUT TO QR (NO PRINT)
84      ;
85      ;
86      000134      RUQN::
87      000134      052767 000000G.000000G. BIS #QN,BASE. ;SET FLAG FOR NO PRINT
88      000142      000404 BR RUNC
89      ;
90      ;
91      ; RUN NORMAL.
92      ; ISSUE QIO FOR UNSOLICITED INTERRUPT FROM TERMINAL.
93      ; (THE LOOP FLAG WILL BE SET). THE LOOP FLAG WILL
94      ; ALLOW CONTINUOUS RUNNING OF THE TERM DETECTOR UNTIL
95      ; A CHARACTER IS ENTERED FROM THE TERMINAL.
96      ;
97      ;
98      ; LOAD BYTE TRANSLATOR. IF A BYTE TRANSLATOR OVERRIDE
99      ; TBAL HAS BEEN BUILT (SEE THE 'CH' AND 'DF' COMMANDS)
100      ; THEN TRANSFER IT TO THE TERM DETECTOR. OTHERWISE TRANSFER
101      ; THE DEFAULT BYTE TRANSLATOR TABLE.
102      ;
103      000144      RUNC::
104      000144      004767 000000G. JSR PC,HANG. ;ALLOW FOR STOPPING TEST.
105      000150      012767 000000 176376 MOV #0,TD$SW. ;RESET TD
106      000156      012702 000004 MOV #4,R2. ;TRANSFER SAME TABLE 4 TIMES
107      000162      016767 000000G.000000G. MOV TLOW,MSTR2. ;LOAD START ADDRESS
108      000170      012700 000100 100$: MOV #64,R0. ;NUMBER OF WORDS IN TABLE
109      000174      032767 000000G.000000G. BIT #RIDE,BASE. ;IS OVERRIDE TABLE IN USE
110      000202      001403 BEQ 12$ ;NO
111      000204      012701 000000G. MOV #BTOVER,R1 ;POINT TO OVERRIDE TABLE
112      000210      000402 BR 14$
113      000212      012701 000000G. 12$: MOV #BTRANS,R1 ;R1 -> BYTE TRANS TABLE
114      000216      012767 000040 176360 14$: MOV #T$ICD,TD$CTW. ;SET INTERFACE AND CONTROL
```

```
115 000224 012767 000030 176362 10$: MOV. #T$BTAR,TD$TAW ;SELECT BYTE TRANS MAR.
116 000232 016767 000000G 176364 MOV. MSTR2,TD$TDW
117 000240 012767 000020 176362 MOV. #T$BT,TD$TAW ;SELECT BYTE TRANS MEMORY.
118 000246 012167 176364 MOV. (R1)+,TD$TDW
119 000252 005267 000000G INC. MSTR2 ;ADVANCE ADDRESS.
120 000256 005300 DEC. R0
121 000260 003361 BGT. 10$
122 000262 005302 DEC. R2 ;SUB FROM OUTER LOOP COUNT.
123 000264 001341 BNE. 100$
124 ;
125 ;
126 ; TRANSFER FROM THE LOCAL DATA BUFFER TO THE TERM DETECTOR
127 ; INPUT BUFFER. THE FIELD 'DATAIN' CONTAINS THE NUMBER OF
128 ; 6-BIT BYTE CHARACTERS IN THE LOCAL BUFFER. CONVERT THIS
129 ; VALUE INTO A NUMBER OF PDP-11 WORDS FOR CONTROLLING THE
130 ; TRANSFER. 4 6-BIT BYTES FIT INTO 3 8-BIT BYTES, 8 6-BIT
131 ; BYTES FIT INTO 3 16-BIT WORDS.
132 000266 016701 000000G MOV. DATAIN,R1 ;NUMBER OF 6-BIT BYTES.
133 000272 006301 ASL. R1 ;MULT BY 2.
134 000274 016702 000000G MOV. DATAIN,R2 ;NUMBER OF 6-BIT BYTES.
135 000300 000201 ADD. R2,R1 ;RESULT = NUMBER OF BYTES X 3
136 000302 006201 ASR. R1 ;NOW DIVIDE BY 8.
137 000304 006201 ASR. R1 ;FOR NUMBER OF
138 000306 006201 ASR. R1 ;WORDS.
139 000310 010146 MOV. R1,-(SP) ;SAVE FOR REPEATED USE.
140 ;
141 ;
142 ; FILL UP THE TERM DETECTOR INPUT BUFFER. MOVE THE
143 ; LOCAL DATA BUFFER TO THE TERM DETECTOR REPEATEDLY
144 ; UNTIL THE TD INPUT BUFFER IS FULL. WRAP AROUND ON BOTH
145 ; BUFFERS. (IE. SINCE THE AMOUNT OF DATA IN THE LOCAL DATA
146 ; BUFFER WILL PROBABLY NOT BE ENOUGH TO FILL THE TD INPUT
147 ; BUFFER, TRANSFER THE SAME DATA REPEATEDLY.
148 000312 011601 MOV. (SP),R1 ;NUMBER OF WORDS IN 'DATA' TABLE.
149 000314 012702 000000G MOV. #DTBL,R2 ;R2 -> DATA TABLE.
150 000320 016703 000000G MOV. ILOW,R3 ;R3 -> INPUT BUFFER.
151 000324 012767 000200 176360 MOV. #T$DISK,TD$CTW ;SIMULATE DISK INPUT.
152 000332 012223 1$: MOV. (R2)+,(R3)+ ;MOVE FROM PGM BUFFER TO TD INPUT BUFFER.
153 000334 016704 176376 MOV. TD$SW,R4 ;LOAD TD STATUS REG.
154 000340 032704 040000 BIT. #T$IBF,R4 ;TEST FOR INPUT BUFFER FULL.
155 000344 001011 BNE. 2$ ;FULL, START CLOCK NOW.
156 000346 020367 000000G CMP. R3,IHIGH ;RUN OUT OF MEMORY
157 000352 003006 BGT. 2$ ;YES, START CLOCK.
158 000354 005301 DEC. R1 ;SUB FROM NUMBER OF WORDS.
159 000356 003365 BGT. 1$ ;MOVE NEXT.
160 000360 011601 MOV. (SP),R1 ;RELOAD NUMBER OF WORDS IN 'DATA' TABLE.
161 000362 012702 000000G MOV. #DTBL,R2 ;R2 -> 'DATA' TABLE.
162 000366 000761 BR. 1$ ;KEEP MOVING.
163 ;
164 ;
165 ; RUN THE TERM DETECTOR. WHEN THE TD INPUT BUFFER GOES
166 ; 'NOT FULL' TRANSFER FROM THE LOCAL DATA BUFFER TO THE TD
167 ; INPUT BUFFER. R2 -> LOCAL DATA BUFFER. R3 -> TD INPUT BUFFER.
168 000370 012767 024200 176360 2$: MOV. #<T$DISK+T$SCLK+T$MODE>,TD$CTW ;SIMULATE DISK INPUT.
169 000376 332767 000000G 000000G 3$: BIT. #LOOP,BASE ;LOOP?.
170 000404 001470 BEQ. RUST ;NO, RESTORE SP AND EXIT.
171 000406 016704 176376 MOV. TD$SW,R4 ;LOAD TD STATUS REG.
```

```
172 000412 032704 040000 BIT #T$IBF,R4 ;IS INPUT BUFFER FULL
173 000416 001367 BNE 3$ ;YES, WAIT FOR NOT FULL
174 000420 020367 000000G CMP R3, IHIGH ;RUN OUT OF INPUT BUFFER
175 000424 003402 BLE 4$ ;NO, CONTINUE
176 000426 016703 000000G MOV ILOW,R3 ;RE-INIT INPUT BUFFER POINTER
177 000432 005301 4$: DEC R1 ;FINISHED WITH 'DATA' BUFFER
178 000434 003003 BGT 5$ ;NO, CONTINUE
179 000436 011601 MOV (SP),R1 ;RELOAD NUMBER OF WORDS IN 'DATA'
180 000440 012702 000000G MOV #DTBL,R2 ;RE-INIT 'DATA' TABLE POINTER
181 000444 012223 5$: MOV (R2)+,(R3)+ ;LOAD INPUT BUFFER
182 ;
183 ; READ TD OUTPUT AVAILABLE REGISTER, ACT DEPENDING ON
184 ; PRINT OPTIONS SET ABOVE
185 ;
186 ; READ WITHOUT PRINT
187 ;
188 000446 032767 000000G 000000G BIT #QN,BASE ;READ TD OUTPUT AND NO PRINT
189 000454 001410 BEQ 7$ ;NO, TRY READ AND PRINT
190 000456 016704 176344 6$: MOV TD$OAR,R4 ;READ OUTPUT AVAILABLE REGISTER
191 000462 032704 100000 BIT #T$OUTA,R4 ;IS OUTPUT AVAILABLE
192 000466 001743 BEQ 3$ ;NO, CONTINUE RUN
193 000470 016705 176346 MOV TD$OTR,R5 ;READ OUTPUT
194 000474 000770 BR 6$ ;READ UNTIL EMPTY
195 ;
196 ; READ AND PRINT
197 ;
198 000476 032767 000000G 000000G 7$: BIT #QP,BASE ;READ OUTPUT AND PRINT
199 000504 001734 BEQ 3$ ;NO, CONTINUE RUN
200 000506 016704 176344 8$: MOV TD$OAR,R4 ;READ OUTPUT AVAILABLE REGISTER
201 000512 032704 100000 BIT #T$OUTA,R4 ;IS OUTPUT AVAILABLE
202 000516 001006 BNE 9$ ;YES, PRINT IT
203 000520 012767 046505 000000G MOV #EM,PRINT ;EMPTY
204 000526 004767 000000G JSR PC,CONSOL ;PRINT MESSAGE
205 000532 000721 BR 3$ ;TRY AGAIN
206 000534 032767 000000G 000000G 9$: BIT #LOOP,BASE ;LOOP ON ?
207 000542 001411 BEQ RUST ;NO, EXIT
208 000544 012705 000000G MOV #PRINT,R5 ;POINT TO PRINT LINE
209 000550 016701 176346 MOV TD$OTR,R1 ;READ OUTPUT
210 000554 004767 000000G JSR PC,UNPK ;CONVERT
211 000560 004767 000000G JSR PC,CONSOL ;PRINT IT
212 000564 000750 BR 8$ ;TRY AGAIN
213 ;
214 000566 005726 RUST: TST (SP)+ ;RESTORE SP
215 000570 RUIX:
216 000570 042767 000000G 000000G BIC #<QP+QN>,BASE ;CLEAR RUN FLAGS
217 000576 004767 000000G JSR PC,KILL ;KILL AST (IF THERE IS ONE)
218 000602 000167 000000G JMP COMXX
219 ;
220 000001 .END
```

```
ALUCKE = 040000
ALUOE = 004000
A01 = 010000
BASE = ***** GX
BITVAL = 000000
BIT0 = 000001
BIT1 = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2 = 000004
BIT3 = 000010
BIT4 = 000020
BIT5 = 000040
BIT6 = 000100
BIT7 = 000200
BIT8 = 000400
BIT9 = 001000
BTOVER = ***** GX
BTRANS = ***** GX
BYTE0 = 000000
BYTE1 = 000001
BYTE10 = 000012
BYTE11 = 000013
BYTE12 = 000014
BYTE13 = 000015
BYTE14 = 000016
BYTE15 = 000017
BYTE16 = 000020
BYTE17 = 000021
BYTE18 = 000022
BYTE19 = 000023
BYTE2 = 000002
BYTE20 = 000024
BYTE21 = 000025
BYTE22 = 000026
BYTE23 = 000027
BYTE24 = 000030
BYTE25 = 000031
BYTE26 = 000032
BYTE27 = 000033
BYTE28 = 000034
BYTE29 = 000035
BYTE3 = 000003
BYTE30 = 000036
BYTE31 = 000037
BYTE32 = 000040
BYTE33 = 000041
BYTE34 = 000042
BYTE35 = 000043
BYTE36 = 000044
BYTE37 = 000045
BYTE38 = 000046
BYTE39 = 000047
BYTE4 = 000004
BYTE40 = 000050
BYTE41 = 000051
BYTE42 = 000052
BYTE43 = 000053
BYTE44 = 000054
BYTE45 = 000055
BYTE46 = 000056
BYTE47 = 000057
BYTE48 = 000060
BYTE49 = 000061
BYTE5 = 000005
BYTE50 = 000062
BYTE51 = 000063
BYTE52 = 000064
BYTE53 = 000065
BYTE54 = 000066
BYTE55 = 000067
BYTE56 = 000070
BYTE57 = 000071
BYTE58 = 000072
BYTE59 = 000073
BYTE6 = 000006
BYTE60 = 000074
BYTE61 = 000075
BYTE62 = 000076
BYTE63 = 000077
BYTE64 = 000100
BYTE65 = 000101
BYTE66 = 000102
BYTE67 = 000103
BYTE68 = 000104
BYTE69 = 000105
BYTE7 = 000007
BYTE70 = 000106
BYTE71 = 000107
BYTE72 = 000110
BYTE73 = 000111
BYTE74 = 000112
BYTE75 = 000113
BYTE76 = 000114
BYTE77 = 000115
BYTE78 = 000116
BYTE79 = 000117
BYTE8 = 000010
BYTE80 = 000120
BYTE81 = 000121
BYTE82 = 000122
BYTE83 = 000123
BYTE84 = 000124
BYTE85 = 000125
BYTE86 = 000126
BYTE87 = 000127
BYTE88 = 000130
BYTE89 = 000131
BYTE9 = 000011
BYTE90 = 000132
BYTE91 = 000133
BYTE92 = 000134
BYTE93 = 000135
BYTE94 = 000136
BYTE95 = 000137
BYTE96 = 000140
BYTE97 = 000141
BYTE98 = 000142
BYTE99 = 000143
BYTVAL = 000144
CBKALL = 001000
CBKCLK = 000400
CNDBRE = 100000
COMXX = ***** GX
CONSOL = ***** GX
CPCCN = 010000
CPREAD = 040000
CPWRTE = 020000
CSADRD = 000004
CSEOC = 100000
CSOE = 000040
CSWRTE = 000100
DATALN = ***** GX
DBR.RD = 000001
DB#CPP = 001457
DB#SPT = 000026
DB#TPC = 000023
DISPGS = 100000
DMAWR = 000005
DMARRD = 000003
DMARWR = 000004
DTBL = ***** GX
ENBR = 010000
ERR18 = ***** GX
ERR3 = ***** GX
FIND = ***** GX
HANG = ***** GX
IHIGH = ***** GX
ILOU = ***** GX
KILL = ***** GX
LOC.EN = 000100
LOC.WA = 040000
LOC.WB = 100000
LOOP = ***** GX
LOOPR = ***** GX
MAREN1 = 000001
MAREN2 = 004000
MARLOD = 010000
MAROUT = 000002
MAR.LO = 002000
MAR.OU = 000400
MBKALL = 001000
MBKCLK = 000400
MMADRD = 000100
MMLEFT = 000002
MMOE = 000004
MMURTE = 000000
MNOBRE = 100000
MREN1 = 000001
MREN2 = 020000
MSTR2 = ***** GX
MSYN = 000040
N = 000144
PLB = 000010
PLC = 000020
PLD = 000030
PLRWR = 000200
PLR.EN = 000200
PRINT = ***** GX
QN = ***** GX
QP = ***** GX
QR#CR1 = 176420
QR#CR2 = 176422
QR#LBR = 176424
QR#ATTN = 000100
QR#CL = 000000
QR#CCCP = 000040
QR#CHB = 000400
QR#CHRL = 000200
QR#CLR = 000040
QR#CNC = 030000
QR#CP = 000060
QR#CPCC = 000010
QR#CP2 = 000260
QR#CSC = 010000
QR#CSEL = 000360
QR#CSET = 000002
QR#CSP = 020000
QDCMA = 000001
QDENBK = 040000
QDENOP = 020000
QFAL = 004000
QFPC = 000045
QFPO = 000044
QFFP = 000046
QHBF = 000002
Q#ICP = 000006
Q#IHB = 000003
Q#IHRL = 000002
Q#IMRP = 000007
Q#LBD = 001000
Q#LBDP = 001001
Q#LBP = 000001
Q#LDCD = 000003
Q#LDMD = 000004
Q#LDPP = 002000
Q#LHP = 010000
Q#MNC = 140000
Q#MR = 000052
Q#MRP = 000040
Q#MRP2 = 000240
Q#MSC = 040000
Q#MSET = 000004
Q#MSP = 100000
Q#NCLK = 176000
Q#PP = 000100
Q#PPSW = 000320
Q#PP2 = 000300
Q#QHLT = 000013
Q#QL = 000043
Q#QLA = 000053
Q#QLB = 000054
Q#QLR = 000001
Q#QW = 000042
Q#RCD = 000005
Q#RDMD = 000006
Q#REBK = 001000
Q#RNC = 006000
Q#RSC = 004000
Q#RESET = 000010
Q#SM = 100000
Q#SP = 000120
Q#SP2 = 000340
RGQ.EN = 000200
RGQ.VA = 020000
RIDE = ***** GX
RUNC = 000144RG 002
RUON = 000134RG 002
RUQP = 000124RG 002
RUSC = 000070RG 002
RUST = 000566R 002
RU1 = 000020RG 002
RU1LN = 000004 G
RU1TBL = 000000RG 002
RU1X = 000570R 002
SCAN = ***** GX
SEQ.CI = 000010
S#CLR = 000000
S#LA = 000001
S#QB = 000005
S#QR = 000006
S#QX = 000004
S#SR = 000007
S#S1 = 000010
S#S2 = 000014
TD#CTR = 176370
TD#CTW = 176360
TD#INL = 004000
TD#MCHP = 000270
TD#OAR = 176344
TD#OTR = 176346
TD#QRD = 000274
TD#SW = 176376
TD#TAR = 176372
TD#TAM = 176362
TD#TDR = 176374
TD#TDW = 176364
TLOW = ***** GX
T#AD = 000020
T#BA = 000002
T#BD = 000010
```

TRUN- M1110 27-MAR-80 15:40 PAGE 5-5
SYMBOL TA

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

T\$OSO = 100000	T\$SCLK = 020000	WORD26 = 000064	WORD51 = 000146	WORD77 = 000232
T\$BT = 000020	T\$SEG1 = 000000	WORD27 = 000066	WORD52 = 000150	WORD78 = 000234
T\$BTAR = 000030	T\$SEG2 = 000001	WORD28 = 000070	WORD53 = 000152	WORD79 = 000236
T\$BTD = 002000	T\$SEG3 = 000002	WORD29 = 000072	WORD54 = 000154	WORD8 = 000020
T\$CD = 000100	T\$SO = 000001	WORD3 = 000006	WORD55 = 000156	WORD80 = 000240
T\$CLK = 002000	T\$UBUS = 100000	WORD30 = 000074	WORD56 = 000160	WORD81 = 000242
T\$DISK = 000200	T\$1CLK = 000400	WORD31 = 000076	WORD57 = 000162	WORD82 = 000244
T\$DRD = 000004	T\$BBEN = 000020	WORD32 = 000100	WORD58 = 000164	WORD83 = 000246
T\$EMEM = 010000	UBD, IN = 000020	WORD33 = 000102	WORD59 = 000166	WORD84 = 000250
T\$FSA = 000000	UNPK = ***** GX	WORD34 = 000104	WORD6 = 000014	WORD85 = 000252
T\$FSAB = 000004	WORD0 = 000000	WORD35 = 000106	WORD60 = 000170	WORD86 = 000254
T\$FSAC = 000014	WORD1 = 000002	WORD36 = 000110	WORD61 = 000172	WORD87 = 000256
T\$FSB2 = 000010	WORD10 = 000024	WORD37 = 000112	WORD62 = 000174	WORD88 = 000260
T\$IB = 000026	WORD11 = 000026	WORD38 = 000114	WORD63 = 000176	WORD89 = 000262
T\$IBAR = 000024	WORD12 = 000030	WORD39 = 000116	WORD64 = 000200	WORD9 = 000022
T\$IBE = 020000	WORD13 = 000032	WORD4 = 000010	WORD65 = 000202	WORD90 = 000264
T\$IBF = 040000	WORD14 = 000034	WORD40 = 000120	WORD66 = 000204	WORD91 = 000266
T\$ICD = 000040	WORD15 = 000036	WORD41 = 000122	WORD67 = 000206	WORD92 = 000270
T\$MODE = 004000	WORD16 = 000040	WORD42 = 000124	WORD68 = 000210	WORD93 = 000272
T\$OB = 000035	WORD17 = 000042	WORD43 = 000126	WORD69 = 000212	WORD94 = 000274
T\$OBE = 004000	WORD18 = 000044	WORD44 = 000130	WORD7 = 000016	WORD95 = 000276
T\$OBF = 010000	WORD19 = 000046	WORD45 = 000132	WORD70 = 000214	WORD96 = 000300
T\$OBRA = 000034	WORD2 = 000004	WORD46 = 000134	WORD71 = 000216	WORD97 = 000302
T\$OBWA = 000032	WORD20 = 000050	WORD47 = 000136	WORD72 = 000220	WORD98 = 000304
T\$OUTA = 100000	WORD21 = 000052	WORD48 = 000140	WORD73 = 000222	WORD99 = 000306
T\$RBD = 000200	WORD22 = 000054	WORD49 = 000142	WORD74 = 000224	WDIVAL = 000310
T\$RNB = 000040	WORD23 = 000056	WORD5 = 000012	WORD75 = 000226	XTREAD = 001000
T\$RSET = 040000	WORD24 = 000060	WORD50 = 000144	WORD76 = 000230	XTWTE = 000400
T\$SC = 000022	WORD25 = 000062			

. ABS. 000000 000
000000 001
TRUN. 000606 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3221 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:45
TRUN, TRUN/-SP=[20,1]IM,[20,1]TRUN

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

AIDTD.TSK MEMORY ALLOCATION MAP. TKB
27-MAR-80

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

TASK. NAME. : ...AID.
PARTITION.NAME.: GEN
IDENTIFICATION.: 08
TASK. UIC. : [20.3].
STACK. LIMITS: 040216 041215 001000 00512.
PRG.XFR.ADDRESS: 051734
TASK.ATTRIBUTES: AL,CP.
TOTAL.ADDRESS.WINDOWS: 2.
TASK. IMAGE. SIZE. : 8032. WORDS.
TASK.ADDRESS.LIMITS: 040000 077253
R-W DISK.BLK.LIMITS: 000042 000110 000047 00039.

AIDTD.TSK:3 OVERLAY DESCRIPTION:

BASE.	TOP.	LENGTH.	
040000	071007	031010	12800.
071010	072077	001070	00560.
071010	073353	002344	01252.
071010	077253	006244	03236.
071010	071617	000610	00392.

TMAIN.
TREG.
TMEH.
TDATA.
TRUN.

*** ROOT SEGMENT: TMAIN.

R/W MEM. LIMITS: 040000 071007 031010 12808.
DISK.BLK.LIMITS: 000042 000073 000032 00026.

MEMORY ALLOCATION SYNOPSIS:

SECTION	TITLE	IDENT	FILE
. BLK: (RW, I, LCL, REL, CON)	041216	021300	08896.
	041216	014224	06292.
\$\$ALER: (RW, I, LCL, REL, CON)	062516	000024	00020.
\$\$ALVC: (RW, D, LCL, REL, CON)	062542	000130	00008.
\$\$AUTO: (RW, I, LCL, REL, CON)	062672	000130	00008.
\$\$FSR1: (RW, D, GBL, REL, OVR)	063022	002040	01056.
	063022	002040	01056.
\$\$FSR2: (RW, D, GBL, REL, CON)	065062	000104	00068.
\$\$MRKS: (RO, I, LCL, REL, OVR)	070540	000076	00062.
\$\$OVDI: (RW, D, LCL, REL, OVR)	065166	000020	00016.
\$\$OVRG: (RW, I, LCL, ABS, CON)	000000	000000	00000.
\$\$RDSG: (RO, I, LCL, REL, OVR)	070636	000150	00104.
\$\$RESL: (RW, I, LCL, REL, CON)	065206	003232	01690.
\$\$RESM: (RW, I, LCL, REL, CON)	132000	007656	04014.
\$\$RGDS: (RW, D, LCL, REL, CON)	070440	000000	00000.
\$\$RTS: (RW, I, GBL, REL, OVR)	070440	000002	00002.
\$\$SGD0: (RW, D, LCL, REL, OVR)	070442	000000	00000.
\$\$SGD1: (RW, D, LCL, REL, CON)	070442	000074	00060.
\$\$SGD2: (RW, D, LCL, REL, OVR)	070536	000002	00002.
\$\$WINDS: (RW, D, LCL, REL, CON)	070540	000000	00000.

GLOBAL SYMBOLS:

AHIGH: 041424-R	CLOW: 041436-R	EFN.1 000001	ERR21 055026-R	INCVAL: 041376-R	OLOW: 041446-R	SCAN: 054224-R
ALOW: 041426-R	CMILUN: 000002	ENDMEM 055002-R	ERR22 055022-R	IO.DET: 002000	ONCE: 000010	SELTST: 055156-R
APLACE: 041234-R	CNUM: 000015	ENDTST 054772-R	ERR23 055016-R	IO.RVB: 010400	OUT: 002000	STAT: 041222-R
ASTFLG: 000020	COMXX: 051740-R	ENFILE 054776-R	ERR3 055136-R	IO.WVB: 011000	OUT1: 055012-R	STOP: 055006-R
BASE: 041232-R	CONSOL: 054610-R	EN1 062632-R	ERR4 055132-R	KILL: 054124-R	PACK: 054434-R	ST1: 062552-R
BASEL: 055162-R	COUNT: 041404-R	ERR1 055146-R	ERR5 055126-R	LDCNTL: 052440-R	PDATA: 052620-R	THIGH: 041450-R
BHIGH: 041430-R	CTBL: 041454-R	ERR10 055102-R	ERR6 055122-R	LD1: 062562-R	PRCNTL: 053032-R	TLOW: 041450-R
BINWD: 041230-R	DATALN: 041406-R	ERR11 055076-R	ERR7 055116-R	LI1: 062642-R	PRDATA: 053126-R	TRFDB: 051536-R
BLOW: 041432-R	DATA1: 041416-R	ERR12 055072-R	ERR8 055112-R	LOOP: 000004	PRINT: 047072-R	TRLUN: 000004
BTEND: 046666-R	DATA2: 041420-R	ERR13 055066-R	ERR9 055106-R	LOOPR: 054004-R	PR1: 062572-R	TR1: 062652-R
BTOVER: 046670-R	DATA3: 041422-R	ERR14 055062-R	FIND: 054316-R	LPTST: 055152-R	QN: 001000	TR6TBL: 046266-R
BTRANS: 046466-R	DA1: 052612-R	ERR15 055056-R	GCMBUF: 041236-R	LUN.TT: 000001	QP: 000400	TXTFDB: 051340-R
BUFFSA: 052044-R	DBLD: 053312-R	ERR16 055052-R	GCMLN: 041360-R	MEND: 041374-R	RE1: 062542-R	TXTLUN: 000003
BUFSET: 052232-R	DF1: 062622-R	ERR17 055046-R	GCMPT: 041362-R	MSTRT: 041370-R	RIDE: 000100	UNPK: 054544-R
BUF62: 052222-R	DPLUS: 041410-R	ERR18 055042-R	GCONLY: 053726-R	MSTR2: 041372-R	RP: 000200	UNP: 041402-R
CHIGH: 041434-R	DSAVE: 043540-R	ERR19 055036-R	HANG: 054040-R	OFFSET: 053224-R	RSPCN: 041400-R	WORDS: 041416-R
CHLEN: 041414-R	DSEND: 046264-R	ERR2: 055142-R	IHIGH: 041440-R	OFF6: 041412-R	RTNPT: 041366-R	\$DIV: 007146
CH1: 062602-R	DTBL: 041540-R	ERR20 055032-R	ILOW: 041442-R	OHIGH: 041444-R	RUH: 062562-R	\$MUL: 007116

AIDTD.TSK:3 MEMORY ALLOCATION MAP TKB
TMEM 27-MAR-88

PAGE 4

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

*** SEGMENT: TMEM ***

MEM. LIMITS: 071010 073353 002344 01252.
TASK-BLK LIMITS: 000076 000100 000003 00003.

MEMORY ALLOCATION SYNOPSIS:

SECTION...	TITLE...	IDENT...	FILE...
. BLK: (RW, I, LCL, REL, CON)	071010	000000	00000.
TMEM: (RW, I, LCL, REL, CON)	071010	002344	01252.
	071010	002344	01252. TMEM
\$\$ALVC: (RW, D, LCL, REL, CON)	073354	000000	00000.
\$\$RTS: (RW, I, GBL, REL, OVR)	070440	000002	00002.

GLOBAL SYMBOLS:

LDAM	071342-R	LDCM	071432-R	LD1	071070-R	PRAM	072660-R	PRCM	072750-R	PR1	072372-R
LDBM	071376-R	LDIB	071466-R	LD1LN	000006	PRBM	072714-R	PRIB	073004-R	PR1LN	000006
LDBT	072122-R	LD0B	071664-R	LD1TBL	071010-R	PRBT	073220-R	PROB	073102-R	PR1TBL	071040-R

*** SEGMENT: TDATA

R/W MEM LIMITS: 071010 077253 006244 03236.
 DISK BLK LIMITS: 000101 000107 000007 00007.

MEMORY ALLOCATION SYNOPSIS:

SECTION	TITLE	IDENT	FILE
. BLK: (RW, I, LCL, REL, CON)	071010	000000	00000.
TDATA: (RW, I, LCL, REL, CON)	071010	002364	01268.
	071010	002364	01268.
\$\$\$ALVC: (RW, D, LCL, REL, CON)	073374	000000	00000.
\$\$\$RESL: (RW, I, LCL, REL, CON)	073374	003660	01968.
\$\$\$RTS: (RW, I, GBL, REL, OVR)	070440	000002	00002.

TDATA: TDATA.OBJ:1

GLOBAL SYMBOLS:

CHBT: 071406-R	CH1TBL: 071010-R	DF1: 072326-R	LIBT: 072102-R	LI1TBL: 071020-R	TR1LN: 000002
CHIB: 071270-R	DA1: 071050-R	DF1LN: 000002	LI1B: 071702-R	TRBT: 073236-R	TR1TBL: 071040-R
CH1: 071160-R	DFBT: 072606-R	DF1TBL: 071030-R	LI1: 071562-R	TR1B: 073144-R	
CH1LN: 000002	DF1B: 072376-R	EN1: 072234-R	LI1LN: 000002	TR1: 073064-R	

AIDTD.TSK:3 MEMORY ALLOCATION MAP: TKB
TRUN: 27-MAR-80

PAGE:6

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

*** SEGMENT: TRUN.

R/W MEM LIMITS: 071010 071617 000610 00392.
DISK BLK LIMITS: 000110 000110 000001 00001.

MEMORY ALLOCATION SYNOPSIS:

SECTION...	TITLE...	IDENT...	FILE...
. BLK: (RW, I, LCL, REL, CON)	071010	000000	00000.
TRUN: (RW, I, LCL, REL, CON)	071010	000606	00390.
	071010	000606	00390. TRUN.
\$\$\$ALVC: (RW, D, LCL, REL, CON)	071616	000000	00000.
\$\$\$RTS: (RW, I, GBL, REL, OVR)	070440	000002	00002.

GLOBAL SYMBOLS:

RUNC- 071154-R RUQN 071144-R RUQP 071134-R RUSC 071100-R RU1 071030-R RU1LN 000004 RU1TBL 071010-R

*** TASK BUILDER STATISTICS:

TOTAL WORK FILE REFERENCES: 40221.
WORK FILE READS: 0.
WORK FILE WRITES: 0.
SIZE OF CORE POOL: 9802, WORDS (38, PAGES)
SIZE OF WORK FILE: 8192, WORDS (32, PAGES)

ELAPSED TIME: 00:00:28.

```

1      .TITLE. QMAIN.
2      ;
3      ;
4      ;
5      ;
6      ;
7      ;
8      ;
9      ;
10     ;
11     ;
12     ;
13     ;
14     ;
15     ;
16     ;
17     ;
18     ;
19     ;
20     ;
21     ;
22     ;
23     ;
24     ;
25     ;
26     ;
27     ;
28     ;
29     ;
30     ;
31     ;
32     ;
33     ;
34     ;
35     ;
36     ;
37     ;
38     ;
39     ;
40     ;
41     ;
42     ;
43     ;
44     ;
45     ;
46     ;
47     ;
48     ;
49     ;
50     ;
51     ;
52     ;
53     ;
54     ;
55     ;
56     ;

```

HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS.
MAIN MODULE.

THIS MODULE PASSES CONTROL TO ITS SUB-MODULES (HQR PROCESSOR
MODULES) BASED ON INFORMATION IN THE COMMAND LINE (TERMINAL
INPUT). THE SUB-MODULES CONTINUE TO PARSE THE COMMAND LINE,
TRANSFERRING CONTROL TO THEIR SUB-MODULES. THESE SUB-MODULES
CONTINUE TO PARSE THE COMMAND LINE, TRANSFERRING CONTROL TO
THEIR SUB-ROUTINES. THE SUB-ROUTINES DO THE ACTUAL INTERFACING
WITH THE HARDWARE. THUS THERE ARE FOUR LEVELS OF CONTROL. FOR
EXAMPLE, TAKE THE COMMAND:
>MR.LD.MD.B
THE 'MR' REPRESENTS THE FIRST LEVEL OF CONTROL AND IS PARSED
BY THE MODULE QMAIN. THE 'LD' REPRESENTS THE SECOND LEVEL OF
CONTROL AND IS PARSED BY THE QMAIN SUB-MODULE MRP. THE 'MD'
REPRESENTS THE THIRD LEVEL OF CONTROL AND IS PARSED BY THE
MRP SUB-MODULE MRLD. THE FOURTH LEVEL OF CONTROL, A SUB-ROUTINE
OF MRLD, ACTUALLY CONTROLS THE LOADING OF THE HARDWARE.

SUB-MODULES OF QMAIN:

MRP. MATCH REPORT PROCESSOR.
CP. CONTROL PROCESSOR.
BCE. BUS CONTROL ELEMENT.
PPS. PIPELINED PROCESSORS.
SP. SUBDOCUMENT PROCESSOR.

SUB-MODULES OF MRP:

MRLD. LOAD
MRPR. PRINT
MPREST. ALL OTHER COMMANDS (EXCEPT MICROCODE DEBUGGING COMMANDS)
MRBUG. DEBUGGING COMMANDS.

SUB-MODULES OF CP:

CPLD. LOAD
CPPR. PRINT
CPREST. ALL OTHER COMMANDS (EXCEPT MICROCODE DEBUGGING COMMANDS)
CPBUG1. SOME MICROCODE DEBUGGING COMMANDS.
CPBUG2. THE REST OF THE DEBUGGING COMMANDS

SUB-MODULES OF BCE:

BCEST. ALL BCE COMMANDS.

SUB-MODULES OF PPS:

PPLD. LOAD
PPPR. PRINT
PPREST. ALL OTHER COMMANDS.

SUB-MODULES OF SP:

SPLD. LOAD
SPPR. PRINT
SPREST. ALL OTHER COMMANDS.

```

58      :      CONTROL IS RETURNED TO QMAIN WHEN ONE OF ITS SUB-MODULES
59      :      ENCOUNTERS AT THE SECOND LEVEL OF CONTROL (SEE ABOVE) A
60      :      STRING THAT IT CANNOT PARSE. THE ASSUMPTION IS THAT THIS
61      :      STRING IS THE MNEMONIC FOR ANOTHER PROCESSOR, REQUIRING A
62      :      TRANSFER OF CONTROL. FOR EXAMPLE, THE COMMAND
63      :      MR>CP LD CS 0
64      :      CANNOT BE PARSED BY MRP ('CP' IS NOT A VALID MRP COMMAND).
65      :      CONTROL IS PASSED TO QMAIN WHICH CAN PARSE 'CP'. CONTROL
66      :      WILL THEN BE TRANSFERRED FROM QMAIN TO THE SUB-MODULE CP.
67      :
68      :
69      :
70      :      QMAIN ALSO CONTAINS DATA AND SUBROUTINES COMMON TO ALL OF ITS
71      :      SUB-MODULES. THERE ARE ALSO MODULES CONTAINING SUBROUTINES
72      :      FOR THE QMAIN SUB-MODULES. THE SUBROUTINE MODULES ARE MRPSUB,
73      :      CPSUB, AND PPSUB. ROUTINES IN THESE MODULES ARE GLOBAL, ALLOW-
74      :      ING CROSS-USAGE. THUS THE QEX WINDOW MEMORY LOADING ROUTINE
75      :      (LD4QW IN SUB-MODULE PPLD OF PPS) CALLS THE SUBROUTINE
76      :      SEQCS IN MODULE CPSUB AND SUBROUTINE SEQM1 IN MODULE MRPSUB.
77      :
78      :
79      :      ASSEMBLY:
80      :      MCR>MAC QMAIN,LP=IMD4,QMAIN.          FROM [5,3].
81      :
82      :      TASK BUILD:
83      :      1. HQR STAND-ALONE PACK COMMAND FILE AIDQ2.CMD
84      :      AIDQ/AL/CP,DA,AIDQ=QMAIN,MRP,MRLD,MRPR,MRREST,MRBUG,
85      :      CP,CPLD,CPPR,CPREST,CBUG1,CBUG2,
86      :      BCE,BCREST,
87      :      PPS,PPLD,PPPR,PPREST,
88      :      MRPSUB,CPSUB,PPSUB,[1.50]RSX11M,STB/SS.
89      :      //
90      :      PAR=PAR14K.
91      :      ASG=TT0:1:2.
92      :      //
93      :
94      :
95      :      2. NPIC SYSTEM COMMAND FILE AIDQ.CMD (USING OVERLAYS)
96      :      ***** NOTE *****
97      :      TASK BUILD MUST BE DONE ON NPIC /04
98      :      MAPPING DOES NOT COME OUT RIGHT WHEN TKB IS DONE ON /05
99      :      EG. ADDRESS OF SYSTEM ROUTINE $DIV COMES OUT INCORRECTLY.
100     :      ***** NOTE *****
101     :
102     :      AIDQ/AL/CP,AIDQ=AIDQ/MP.
103     :      TASK=AIDQ.
104     :      PAR=GEN:40000:40000
105     :      ASG=TT0:1:2.
106     :      //
107     :
108     :      OVERLAY DESCRIPTION AIDQ.ODL:
109     :      .ROOT. QMAIN-[1.50]RSX11M,STB/SS-*(A,B,C,D)
110     :      A:      .FCTR MRP-MRPSUB-CPSUB-*(MRLD,MRPR,MRREST,MRBUG)
111     :      B:      .FCTR CP-MRPSUB-CPSUB-PPSUB-(CBUG1,CBUG2,CPREST,CPLD,CPPR)
112     :      C:      .FCTR BCE-BCREST.
113     :      D:      .FCTR PPS-MRPSUB-CPSUB-PPSUB-(PPLD,PPPR,PPREST)
114     :      .END.

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

115      ;
116      ;
117      ;
118      ;
119      ;
120      ;
121      ;
122      ;
123      ;
124      ;
125      ;
126      ;
127      ;
128      ;
129      ;
130      ;
131      ;
132      ;
133      ;
134      ;
135      ;
136      ;
137      ;
138      ;
139      ;
140      ;
141      ;
142      ;
143      ;
144      ;
145      ;
146      ;
147      ;
148      ;
149      ;
150      ;
151      ;
152      ;
153      ;
154      ;
155      ;
156      ;
157      ;
158      ;
159      ;
160      ;
161      ;
162      ;
163      ;
164      ;
165      ;
166      ;
167      ;
168      ;
169      ;
170      ;
171      ;

LOCAL DATA AREAS

.MCALL QIOW$,QIO$,EXIT$,ABRT$,GCML$,GCMLB$,FSRSZ$,CLEF$,
.MCALL ASTX$,RDAF$,WTSE$,SETF$,RQST$

.GLOBAL IO,WVB,IO,RVB,IO,ATA,IO,DET

G.DPRM == 000160
LUN.TT == 1
EFN.1 == 1
EFN.2 == 2
EFN.3 == 3
EFN.4 == 4
EFN.33 == 33
CMILUN == 2

LOOP == 4
ONCE == 100
ASTFLG == 200
RP == 1000
OUT == 2000
BREAK == 4000
NEQLB == 10000

;READ/WRITE TT0
;EVENT FLAG FOR TT0
;EVENT FLAG FOR HQR INTERRUPTS
;EVENT FLAG FOR UNSOLICITED TERMINAL INTERRUPTS
;EVENT FLAG FOR COMMUNICATION WITH HQR LOADER

;LOOP ON COMMAND
;PRINT ONE WORD ONLY
;QIO + AST ISSUED
;REPEAT PROMPT
;CONTROL PRINTING OF MEMORY CONTENTS
;BREAKPOINT SET
;NO QLB ERASE

.NLIST BEX
MYSELF:: .RAD50 /AIDQR/
TSKTCB:: .WORD 0
OLDVEC:: .WORD 0
LOADER:: .RAD50 /LOADER/
EFBUF:: .BLKW 4
STAT:: .BLKW 2
ERWORD:: .WORD 0
BINWD:: .WORD 0
BASE:: .WORD 0
APLACE:: .WORD 0
GCMBUF:: .BLKW 41
GCMLN:: .WORD 0
GCMPNT:: .WORD 0
ASTWRD:: .WORD 0
RTNPT:: .WORD 0
MSTR1:: .WORD 0
MSTR2:: .WORD 0
MEND:: .WORD 0
INCVAL:: .WORD 0
RSPCNT:: .WORD 0
UPLIM:: .WORD 0

;TCB OF MY TASK
;OLD VECTOR AT 274
;EVENT FLAG BUFFER
;INDEX VALUE FOR ERROR MESSAGE TABLE
;TARGET FOR NUMERIC CONVERSIONS FROM ASCII
;ALL PURPOSE FLAG
;PRELIM BIT SETTINGS FOR CSR 1
;COMMAND LINE BUFFER
;COMMAND LINE LENGTH
;COMMAND LINE POINTER
;RECEIVER FOR AST CHAR
;RTN ADDR SAVE AREA
;START ADDR FOR MEMORY LOADING/PRINTING
;WORKING ADDR FOR LOAD/PRINT
;END ADDR FOR MEMORY LOADING/PRINTING
;MEMORY INCREMENT VALUE
;(COMMAND LINE) RESPONSE COUNT
;MEMORY UPPER LIMIT

WJORDS:
DATA1:: .WORD 0
DATA2:: .WORD 0
DATA3:: .WORD 0
DATA4:: .WORD 0

;LOAD/PRINT VALUES

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

172.                                     :      MEMORY LIMITS TABLE
173.                                     :
174 000216 000377 MMHIGH:: .WORD 255. :MRP MICROPGM MEMORY
175 000220 000000 MMLOW:: .WORD 0 :
176 000222 007777 MDHIGH:: .WORD 4095. :MRP DATA MEMORY
177 000224 000000 MDLOW:: .WORD 0 :
178 000226 001777 CSHIGH:: .WORD 1023. :CP CONTROL STORE
179 000230 000000 CSLOW:: .WORD 0 :
180 000232 007777 CDHIGH:: .WORD 4095. :CP DATA MEMORY
181 000234 000000 CDLOW:: .WORD 0 :
182 000236 077777 QXHIGH:: .WORD 077777 :X'7FFF' QEX MEMORIES
183 000240 076000 QXLOW:: .WORD 076000 :X'7C00'
184 000242 007777 FAHIGH:: .WORD 4095. :FAL MEMORIES
185 000244 000000 FALOW:: .WORD 0 :
186 000246 002000 LHHIGH:: .WORD 1024. :LHP MEMORIES (ALLOW X'400' ILLEGAL ADDRESS)
187 000250 000000 LHLOW:: .WORD 0 :
188 000252 000177 HLHIGH:: .WORD 127. :HRL BUFFER IN CC MEMORY
189 000254 000000 HLLOW:: .WORD 0 :
190 000256 000014 BLHIGH:: .WORD 12. :BCL IN CC MEMORY
191 000260 000000 BLLOW:: .WORD 0 :
192 000262 000377 SQHIGH:: .WORD 255. :SP REFERENCE, QLB MEMORIES
193 000264 000000 SQLOW:: .WORD 0 :
194 000266 000377 SRHIGH:: .WORD 255. :SP SIDREAD MEMORY
195 000270 000000 SRLOW:: .WORD 0 :
196 000272 007777 SDHIGH:: .WORD 4095. :SP SIDMEM MEMORY
197 000274 000000 SDLOW:: .WORD 0 :
198.                                     :
199.                                     :
200.                                     :      TABLE USED IN CONVERSION FROM ASCII HEX TO BINARY
201 000276 TRTBL::
202.                                     :
203 000356 000 001 002. .+60
204 000377 000 001 002. .BYTE 0,1,2,3,4,5,6,7,8,9.
205 000377 012 013 014. . TRTBL+101
206 000475 000 013 014. .BYTE 10,11,12,13,14,15.
207.                                     :
208.                                     :
209.                                     :      TBAL USED IN CONVERSION FROM BINARY TO ASCII HEX
210 000475 060 061 062. TRTBL2:: .ASCII /0123456789ABCDEF/
211.                                     :

```



```

213
214
215
216
217
218
219
220 000516
221 000516      115      122
222 000520 000000G
223 000522      103      120
224 000524 000000G
225 000526      102
226 000530 000000G
227 000532      120      120
228 000534 000000G
229 000536      123      120
230 000540 000000G
231 000542      105      130
232 000544 002370
233 000006
234
235
236
237
238 000546      015      012
239 000550
240      000116
241
242
243
244
245
246
247
248 000666      000
249 000667      015      012      015
250 000673      105      130      111
251 000722      015      012      000
252 000725      015      012
253 000727      124      111      115
254 000754      015      012
255 000756      105      116      104
256 001002      015      012
257 001004      105      116      104
258 001032      015      012
259 001034      105      116      124
260 001075      015      012      015
261 001103      110      101      122
262 001160      015      012      000
263 001163      015      012
264 001165      111      116      126
265 001205      015      012
266 001207      111      116      126
267 001233      015      012
268 001235      111      116      126
269 001271      015      012

```

;
 ;
 ; FIRST LEVEL CONTROL TABLE
 ; PROCESSOR MNEMONICS AND ASSOCIATED QMAIN SUB-MODULE
 ; ADDRESSES
 ;
 ;
 FTBL:
 .ASCII /MR/ ;MATCH REPORT PROCESSOR
 .WORD MRP
 .ASCII /CP/ ;CONTROL PROCESSOR
 .WORD CP
 .ASCII /BC/ ;BUS CONTROL ELEMENT
 .WORD BCE
 .ASCII /PP/ ;PIPELINE PROCESSORS
 .WORD PPS
 .ASCII /SP/ ;SUBDOCUMENT PROCESSOR
 .WORD SPS
 .ASCII /EX/ ;EXIT
 .WORD EXIT
 FNUM = <.-FTBL>/4
 ;
 ;
 ; PRINT LINE
 ;
 ;
 PRINT:
 .BYTE 15,12 ;PRECEDENCE PRINT LINE WITH CRLF
 .REPT 78
 .BYTE 40
 .ENDR
 ;
 ;
 ; TABLE OF MESSAGES
 ;
 ;
 .BYTE 0
 .BYTE 15,12,15,12
 .ASCII /EXIT-HQR DEBUGGING AIDS/
 .BYTE 15,12,0
 .BYTE 15,12
 .ASCII /TIME-OUT-ON-FREE-RUN/
 .BYTE 15,12
 .ASCII /END OF FILE REACHED/
 .BYTE 15,12
 .ASCII /END OF MEMORY REACHED/
 .BYTE 15,12
 .ASCII /ENTER ANY CHARACTER TO EXIT LOOP/
 .BYTE 15,12,15,12,15,12
 .ASCII /HARDWARE QUERY RESOLVER MANUAL DEBUGGING AIDS/
 .BYTE 15,12,0
 .BYTE 15,12
 .ASCII /INVALID COMMAND/
 .BYTE 15,12
 .ASCII /INVALID LOOP OPTION/
 .BYTE 15,12
 .ASCII /INVALID UPPER MEMORY LIMITS/
 .BYTE 15,12

270 001273	111	116	126	.ASCIZ	/INVALID LOWER MEMORY LIMITS/
271 001327	015	012		.BYTE	15,12
272 001331	111	116	103	.ASCIZ	/INCORRECT COMMAND CHARACTER COUNT/
273 001373	015	012		.BYTE	15,12
274 001375	111	116	126	.ASCIZ	/INVALID MEMORY MNEMONIC/
275 001425	015	012		.BYTE	15,12
276 001427	111	116	126	.ASCIZ	/INVALID REGISTER MNEMONIC/
277 001461	015	012		.BYTE	15,12
278 001463	111	116	126	.ASCIZ	/INVALID NUMERIC VALUE/
279 001511	015	012		.BYTE	15,12
280 001513	115	111	123	.ASCIZ	/MISSING OPERAND/
281 001533	015	012		.BYTE	15,12
282 001535	115	111	123	.ASCIZ	/MISSING COMMAND/
283 001555	015	012		.BYTE	15,12
284 001557	116	117	116	.ASCIZ	/NON-EXISTENT PROCESSOR/
285 001606	015	012		.BYTE	15,12
286 001610	120	122	117	.ASCIZ	/PROCESSOR MNEMONIC MUST PRECEED NEXT COMMAND/
287 001665	377			ASCIZ::	.BYTE 377
288				.EVEN	
289				.LIST	BEX
290				.NLIST	CND
291				:	
292				:	
293				:	COMMAND LINE MACRO
294				:	
295				:	
296 001666				GCMBLK::	GCLMB\$ 2,,GCMBUF,CMILUN
297 002174				FERSZ\$	1

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

299      :
300      :
301      : ENTER HERE
302      :
303      :
304 002174      : START:
305 002174      :
306 002200 016767 000000G 175576      : CALL OUT1 ;ISSUE INFORMATION MESSAGE
307 002206 013767 000274 175572      : MOV $TKTCB,TSKTCB ;SAVE MY TCB
308 002214 012737 003330 000274      : MOV @#274,OLDVEC ;SAVE VECTOR AT 274
309      : MOV #BPTISR,@#274 ;MOVE IN MY INTERRUPT HANDLER ADDR
310      :
311      : START OFF WITH MASTER RESET, NO-CLOCKS IN CSR #1
312 002222 012746 177777      : MOV #177777,-(SP) ;CLEAR CSR1
313 002226 012746 000010      : MOV #0$RSET,-(SP) ;SET RESET
314 002232      : CALL CSR1 ;RESET HOR
315 002236 012746 000010      : MOV #0$RSET,-(SP) ;CLEAR RESET
316 002242 012746 176000      : MOV #<Q$NCLK>,-(SP) ;SET NO CLKS
317 002246      : CALL CSR1 ;MOVE TO CSR1
318 002252 012746 000040      : MOV #Q$CLR,-(SP) ;REINHIBIT FAL PROCEESOR
319 002256      : CALL PPCR
320 002262 012746 000000      : MOV #S$CLR,-(SP) ;CLEAR SP ADDRESS SELECT
321 002266      : CALL SPCR
322      :
323      : PUT OUT INITIAL PROMPT. LOCATE 2-CHAR PROCESSOR/COMMAND MNEMONIC.
324      : EG. IF THE COMMAND LINE READS:
325      : >MR LD MD 0
326      : LOCATE THE 'MR'
327      :
328      : MR MATCH REPORT PROCESSOR
329      : CP CONTROL PROCESSOR
330      : BC BUS CONTROL ELEMENT
331      : PP PIPELINED PROCESSORS
332      : SP SUBDOCUMENT PROCESSOR
333      : EX EXIT DEBUGGING AIDS (COMMAND)
334      :
335 002272      : RESEL:
336 002272 012767 020040 177550      : MOV #20040,GCMBLK+G,DPRM+2 ;CLEAR PROCESSOR NAME
337 002300      : CALL GCONLY ;ISSUE GCML
338 002304      : CALL FIND ;FIND THE PROCESSOR MNEMONIC
339 002310 103003      : BCC 1$ ;OK, CONTINUE
340 002312      : CALL ERR2 ;NOTHING IN COMMAND LINE
341 002316 000765      : BR RESEL
342 002320 022700 000002      : CMP #2,R0 ;COMMANDS ARE 2 CHARS
343 002324 001403      : BEQ COMXX
344 002326      : CALL ERR8 ;INCORRECT CHAR COUNT
345 002332 000757      : BR RESEL ;TRY AGAIN
346      :
347      :
348      : TOP OF PROCESSOR LOOP
349      : MATCH PROCESSOR NAME AGAINST TABLE OF NAMES + CONTROL ROUTINE
350      : ADDRESSES
351      :
352      : R1 -> PROCESSOR MNEMONIC IN THE COMMAND LINE
353      :
354      :
355 002334 012700 000006      : COMXX: MOV #FNUM,R0 ;R0 = NUMBER OF PROCESSORS

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

356 002340 012702 000516'      MOV.    #FTBL,R2      ;R2 -> TABLE OF PROCESSOR MNEMONICS.
357 002344                      CALL.    SCAN      ;FIND MATCH IN TABLE
358 002350 103005                BCC.     3$        ;OK, CONTINUE
359 002352                      CALL.    ERR2      ;COMMAND NOT IN TABLE
360 002356                      CALL.    ERR1      ;WHAT TO DO NEXT
361 002362 000743                BR       RESEL     ;TRY AGAIN
362.                             ;
363.                             ;
364.                             ;
365.                             ;
366.                             ;
367.                             ;
368.                             ;
369.                             ;
370 002364 000171 000000.        3$:    JMP.     @R1
371.                             ;
372.                             ;
373.                             ;
374.                             ;
375.                             ;
376 002370                      EXIT:
377 002370 016737 175412 000274  MOV.     OLDVEC,@#274      ;RESTORE ORIGINAL VECTOR CONTENTS
378 002376                      CALL.    ENDTST     ;PUT OUT END OF RUN MESSAGE
379 002402                      EXIT$S

```

```

381      ;
382      ;
383      ;
384      ;
385      ;
386      ;
387      ;
388      ;
389      ;
390      ;
391      ;
392      ;
393      ;
394      ;
395 002410      CSR1::
396 002410 016767 176420 175420      MOV.   QR$CR1, APLACE.      ;GET THE CURRENT VALUE.
397 002416 046667 000004 175412      BIC.   4(SP), APLACE.      ;CLEAR FIRST.
398 002424 056667 000002 175404      BIS.   2(SP), APLACE.      ;THEN SET.
399 002432 016767 175400 176420      MOV.   APLACE, QR$CR1      ;NOW RETURN IT.
400 002440 011666 000004                MOV.   (SP), 4(SP)      ;MOVE RETURN ADDR TO TOP OF STACK.
401 002444 022626                CMP.   (SP)+, (SP)+      ;BUMP STACK POINTER PAST ARGS.
402 002446                RETURN.      ;LEAVE.

```

```
404      ;
405      ;
406      CALL HARDWARE QUERY RESOLVER LOADER
407      ;
408      ; ***** NOTE *****
409      ; WHEN ON THE NPIC SYSTEM DO NOT USE CL IF CCIN AND CCOUT
410      ; ARE RUNNING
411      ; ***** NOTE *****
412      ;
413      ;
414      ; CLEAR GLOBAL EVENT FLAG
415      ;
416 002450 CL::
417 002450 CLEF$S #EFN.33
418      ;
419      ; REQUEST LOADER
420      ;
421 002462 ROST$S #LOADER
422      ;
423      ; WAIT FOR 'LOADER' TO SET EVENT FLAG
424      ;
425 002524 WTSE$S #EFN.33
426 002536 RETURN
```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

428.      ;
429.      ;
430.      ;
431.      ;
432.      ;
433.      ;
434.      ;
435.      ;
436.      ;
437.      ;
438.      ;
439.      ;
440.      ;
441.      ;
442.      ;
443.      ;
444.      ;
445.      ;
446.      ;
447.      ;
448.      ;
449.      ;
450.      ;
451.      ;
452.      ;
453.      ;
454.      ;
455.      ;
456.      ;
457.      ;
458.      ;
459.      ;
460.      ;
461.      ;
462.      ;
463.      ;
464.      ;
465.      ;
466.      ;
467.      ;
468.      ;
469.      ;
470.      ;
471.      ;
472. 002540 012767 000004 175434 BUF54: MOV. #4.RSPCNT. ;NUMBER OF WORDS TO PROMPT.
473 002546 012767 000001 175424 BUF54: MOV. #1.INCVAL. ;BUFFERS INC BY 1
474 002554 000433          BUF54: BR. SET.
475 002556 012767 000003 175416 BUF53: MOV. #3.RSPCNT. ;NUMBER OF WORDS TO PROMPT.
476 002564 012767 000001 175406 BUF53: MOV. #1.INCVAL. ;BUFFERS INC BY 1
477 002572 000424          BUF53: BR. SET.
478 002574 012767 000002 175400 BUF52: MOV. #2.RSPCNT. ;PROMPT 2 WORDS.
479 002602 012767 000001 175370 BUF52: MOV. #1.INCVAL. ;BUFFERS INC BY 1
480 002610 000415          BUF52: BR. SET.
481 002612 012767 000001 175362 BUF51: MOV. #1.RSPCNT. ;PROMPT 1 WORD
482 002620 012767 000001 175352 BUF51: MOV. #1.INCVAL. ;BUFFERS INC BY 1
483 002626 000406          BUF51: BR. SET.
484 002630 012767 000001 175344 BUF5M: MOV. #1.RSPCNT. ;PROMPT 1 WORD

```

SET-UP BEFORE PROMPTING ON ANY 'LD' COMMAND OR BEFORE
 EXECUTING ANY 'PR' COMMAND.
 CALLED BY ROUTINES IN MMLD, MMR, CPLD, CPPR, PLLD.
 PPPR: THE FIELDS SET-UP HERE ARE USED IN THE QMAIN.
 PROMPTING SUBROUTINE PDATA AND THE PRINT ROUTINE PRDATA.

INPUT:
 2(SP) LOAD/PRINT START ADDRESS
 4(SP) LOWER MEMORY LIMIT
 6(SP) UPPER MEMORY LIMIT

OUTPUT:
 INCVAL - VALUE BY WHICH MEMORY ADDRESS INCREMENTS
 MEND - ADDRESS OF LAST WORD IN MEMORY TO BE LOADED/PRINTED
 RSPCNT - NUMBER OF WORDS TO EXPECT IN COMMAND LINE (MEMORY WIDTH)

C-BIT CLEAR: NO ERROR (START ADDRESS IS IN RANGE)
 C-BIT SET: ERROR IN RANGE

MEMORY	INCVAL	RSPCNT
MM	1	2
MD	1	1
CS	1	4
CD	1	1
HL	2	1
BL	2	1
QW	1	1
QL	1	1
FP	1	1
FC	1	1
QR	1	1
Q0	1	1
Q1	1	1
Q2	1	1
QX	1	1
QB	1	1
SR	1	1
S1	1	3
S2	1	3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

485 002636 012767 000002 175334      MOV.    #2, INCVAL.      :BUFFERS INC BY 2.
486                                     ;
487 002644 022767 177777 175324 SET:    CMP.    #-1, MEND.      :MEMORY END ADDR SET UP.
488 002652 001003                                     BNE.    10$              :YES.
489 002654 016667 000004 175314      MOV.    4(SP), MEND.    :SET END ADDR = MEMORY UPPER LIMIT.
490 002662 026766 175304 000002 10$:    CMP.    MSTRT, 2(SP)    :IS LOAD ADDRESS IN RANGE (LOW)
491 002670 103003                                     BHS.    1$              :OK, CONTINUE.
492 002672                                     CALL.   ERR9            :OUT OF RANGE.
493 002676 000424                                     BR.     BUFCX2.
494 002700 026766 175272 000004 1$:    CMP.    MEND, 4(SP)    :IS ADDR IN RANGE (HIGH)
495 002706 101403                                     BLOS.   2$              :YES, CONTINUE.
496 002710                                     CALL.   ERR10           :OUT OF RANGE.
497 002714 000415                                     BR.     BUFCX2.
498 002716 026767 175250 175252 2$:    CMP.    MSTRT, MEND.    :IS START ADDR LOWER THAN END ADDR.
499 002724 101403                                     BLOS.   3$              :YES.
500 002726                                     CALL.   ERR10           :OUT OF RANGE.
501 002732 000406                                     BR.     BUFCX2.
502                                     ;
503 002734 011666 000004 3$:    MOV.    (SP), 4(SP)    :MOVE RETURN ADDRESS
504 002740 062706 000004                                     ADD.    #4, SP.         :ADJUST SP (FOR MEM LIMITS)
505 002744 000241                                     CLC.
506 002746 000405                                     BR.     BUFCX2.
507 002750 011666 000004      BUFCX2: MOV.    (SP), 4(SP)    :RETURN.
508 002754 062706 000004                                     ADD.    #4, SP.         :MOVE RETURN ADDRESS
509 002760 000261                                     SEC.
510 002762      BUFCX2: RETURN.

```


Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

512.      ;
513.      ;
514.      ;
515.      ; PROMPTING CONTROL FOR LOADING ALL MEMORIES AND BUFFERS.
516.      ; READ NUMERIC DATA FROM THE COMMAND LINE, CONVERT AND STORE
517.      ; INTO A COMMON BUFFER, EG. IF THE INITIATING COMMAND IS:
518.      ; CP>LD,CS,0
519.      ; THIS ROUTINE WILL PUT OUT A PROMPT AND EXPECT 4 NUMERIC
520.      ; VALUES IN RETURN:
521.      ; >0000 0000 0000 0000
522.      ; THIS ROUTINE CONVERTS THESE ASCII-HEX VALUES INTO BINARY
523.      ; AND STORES THEM INTO DATA1, DATA2, DATA3, DATA4
524.      ;
525.      ; INPUT: (SET UP BY QMAIN SUBROUTINE BUFSET)
526.      ; MSTR2 - CURRENT MEMORY ADDRESS
527.      ; MEND - MEMORY UPPER ADDRESS LIMIT
528.      ; RSPCNT - NUMBER OF WORDS TO EXPECT IN COMMAND LINE
529.      ;
530.      ; OUTPUT:
531.      ; UWORDS (DATA1, DATA2, DATA3, DATA4) DEPENDING UPON RSPCNT.
532.      ;
533.      ; C-BIT CLEAR, V-BIT CLEAR.      NORMAL RETURN.
534.      ; C-BIT CLEAR, V-BIT SET        <CR> RESPONSE TO PROMPT
535.      ; C-BIT SET, V-BIT CLEAR        END OF MEMORY OR CONVERSION ERROR.
536.      ;
537.      ; REGISTERS 1, 4, 5 DESTROYED.
538.      ;
539.      ; PDATA:
540.      ; 002764      026767      175204      175204      CMP      MSTR2,MEND      ;UPPER MEMORY LIMIT REACHED
541.      ; 002772      101403      ;BLOS      10$      ;NO, CONTINUE.
542.      ; 002774      ;CALL      ENMEM      ;END OF MEMORY REACHED.
543.      ; 003000      000460      ;BR      PDCX      ;SET CARRY AND EXIT.
544.      ;
545.      ; 003002      016701      175166      10$:      MOV      MSTR2,R1      ;PREPARE TO PRINT ADDRESS.
546.      ; 003006      012705      000550      ;MOV      #PRINT,R5      ;POINT TO PRINT LINE
547.      ; 003012      ;CALL      UNPK      ;CONVERT ADDRESS.
548.      ; 003016      ;CALL      CONSOL      ;PRINT OUT ADDRESS.
549.      ;
550.      ; 003022      012767      020040      177016      MOV      #20040,GCMBLK+G,DPRM.      ;ERASE CR AND LF.
551.      ; 003030      012767      020040      177012      MOV      #20040,GCMBLK+G,DPRM+2.      ;ERASE PROCESSOR MNEMONIC.
552.      ; 003036      ;CALL      GCONLY      ;PROMPT.
553.      ; 003042      112767      000015      176776      MOV      #15,GCMBLK+G,DPRM.      ;RESTORE CR.
554.      ; 003050      112767      000012      176771      MOV      #12,GCMBLK+G,DPRM+1      ;RESTORE LF.
555.      ; 003056      ;CALL      FIND      ;LOCATE FIRST DATA WORD IN COMMAND LINE.
556.      ; 003062      103424      ;BCS      PDVX      ;<CR> RESPONSE, EXIT
557.      ;
558.      ; 003064      012705      000206      ;MOV      #UWORDS,R5      ;WORDS FROM COMMAND LINE GO INTO THIS TABLE.
559.      ; 003070      016704      175106      ;MOV      RSPCNT,R4      ;NUMBER OF WORDS TO EXPECT.
560.      ; 003074      ;CALL      PACK      ;CONVERT WORD TO BINARY.
561.      ; 003100      103003      2$:      BCC      3$      ;OK, CONTINUE.
562.      ; 003102      ;CALL      ERR4      ;INVALID NUMERIC VALUE.
563.      ; 003106      000415      ;BR      PDCX      ;AND EXIT.
564.      ; 003110      016725      174716      3$:      MOV      BINWD,(R5)+      ;MOVE WORD TO TABLE.
565.      ; 003114      005304      ;DEC      R4      ;SUB FROM LOOP COUNT.
566.      ; 003116      001414      ;BEQ      PDCX      ;IF ZERO.
567.      ; 003120      ;CALL      FIND      ;FIND NEXT WORD.
568.      ; 003124      103363      ;BCC      2$      ;OK, CONTINUE.

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

QMAIN: M 00 M1110 27-MAR-80 15:17 PAGE 13-1

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

569 003126
570 003132 000403
571
572 003134 000241
573 003136 000262
574 003140 000405
575 003142 000242
576 003144 000261
577 003146 000402
578 003150 000241
579 003152 000242
580 003154

CALL ERR3
BR PDCX
PDVX: CLC
SEV
BR PDDX
PDCX: CLV
SEC
BR PDDX
PDCCX: CLC
CLV
PDDX: RETURN

: *MISSING OPERAND:

: INDICATE <CR> RESPONSE.

: END OF MEMORY OR CONVERSION ERROR.

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

582.      ;
583      ;
584      ;
585      ;      COMMON PRINT ROUTINE.
586      ;      THIS ROUTINE CONVERTS VALUES IN THE FIELD 'UWORDS' TO
587      ;      ASCII HEX (DEPENDING ON RSPCNT) AND PRINTS THEM (TT0)
588      ;
589      ;      INPUT:
590      ;      #OUT FLAG.      - WHEN LOOP OPTION IS IN EFFECT, PRINT 1 WORD ONLY.
591      ;      #ONCE FLAG.     - CONTROL PRINTING OF 1 WORD WHEN #OUT IS SET.
592      ;      RSPCNT.         - NUMBER OF WORDS TO PRINT (SEE 'BUFSET' RTN)
593      ;      MSTR2.          - MEMORY ADDRESS (SEE 'BUFSET')
594      ;      UWORDS.         - MEMORY WORDS (DATA1, DATA2, DATA3, DATA4)
595      ;
596      ;      REGISTERS 1, 3, 4, 5 DESTROYED.
597      ;
598 003156      ;
599 003156      ;      PRDATA::
600 003164      ;      BIT.      #OUT,BASE.      ;OUTPUT CONTROL ON.
601 003166      ;      BEQ.      10$          ;NO, SKIP OUTPUT CONTROL.
602 003174      ;      BIT.      #ONCE,BASE.     ;ONE LINE PRINTED.
603 003176      ;      BNE.      PRDX          ;YES, EXIT
604      ;      B13.      #ONCE,BASE.
605 003204      ;
606 003210      ;      10$:      MOV.      MSTR2,R1      ;FIRST CONVERT ADDRESS TO ASCII.
607 003214      ;      MOV.      #PRINT,R5      ;POINT TO PRINT LINE
608      ;      CALL.      UNPK          ;PERFORM CONVERSION.
609 003220      ;
610 003224      ;      MOV.      #UWORDS,R4      ;POINT TO WORDS FOR PRINTING.
611 003230      ;      MOV.      RSPCNT,R3      ;NUMBER OF MEM WORDS TO PRINT.
612 003234      ;      ADD.      #2,R5          ;ADVANCE PRINT LINE POINTER.
613 003236      ;      MOV.      (R4)+,R1      ;LOAD ONE WORD.
614 003242      ;      CALL.      UNPK          ;CONVERT IT TO ASCII
615 003244      ;      DEC.      R3            ;DEC WORD COUNT.
616 003246      ;      BNE.      1$            ;REPEAT.
617      ;      CALL.      CONSOL.          ;PRINT LINE.
618 003252      ;      PRDX:      RETURN.

```

```

620      ;
621      ;
622      ;
623      ;
624      ;
625      ;
626      ;
627      ;
628      ;
629      ;
630      ;
631      ;
632      ;
633      ;
634      ;
635      ;
636      ;
637      ;
638      ;
639      ;
640      ;
641      ;
642      ;
643      ;

```

GET COMMAND LINE FROM TERMINAL.

OUTPUT:

GCMBUF - WORK AREA TO HOLD COMMAND LINE.

GCMLN - LENGTH OF LINE READ.

GCMPNT - POINTER TO COMMAND LINE (SET TO POINT TO BEGINNING)

SEE 'FIND' SUBROUTINE FOR THE USE AND UPDATING OF THESE FIELDS.

GCONLY:

MOV.	#GCMBUF,R0	:POINT TO COMMAND LINE BUFFER.
MOV.	#41,R1	:NUMBER OF WORDS IN BUFFER.
CLR.	(R0)+	:CLEAR COMMAND LINE.
DEC.	R1	:FINISHED?
BNE.	1\$:NO.

1\$:

GCML\$ #GCMLK.

MOV. G,CMLD(R0),GCMLN.

MOV. #GCMBUF,GCMPNT.

MOV. #GCMBUF,R1.

RETURN.

:SAVE LENGTH.

:INIT COMMAND LINE POINTER.

:POINT R1 TO COMMAND LINE.

Approved For Release 2005/07/17 : CIA-RDP85-00514R000200020001-3

```

666      ;
667      ;
668      ;
669      ;
670      ;
671 003404      LOOPR::
672 003404      CALL    FIND          ;LOCATE RESPONSE
673 003410 103437      BCS    LOOPX          ;NOTHING THERE, EXIT
674 003412 122711 000114      CMPB   #'L.(R1)      ;LOOP INDICATOR
675 003416 001403      BEQ    HANG          ;OK, CONTINUE
676 003420      CALL    ERR11          ;WRONG CHARACTER
677 003424 000431      BR     LOOPX
678      ;
679      ;
680      ;
681      ;
682 003426 052767 000200 174400      HANG:: BIS    #ASTFLG,BASE      ;SET FLAG FOR QIO ISSUED
683 003434 052767 000004 174372      BIS    #LOOP,BASE      ;SET FLAG FOR LOOP
684 003442      CALL    STOP          ;GIVE DIRECTIONS FOR STOPPING TEST
685      ;
686 003446      QIO$S   #IO.ATA,#LUN.TT,....,<#AST>
687      ;
688 003510      LOOPX:
689 003510      RETURN
690      ;
691      ;
692      ;
693      ;
694      ;
695 003512      HANG2::
696 003512 052767 000200 174314      BIS    #ASTFLG,BASE      ;SET FLAG FOR QIO ISSUED
697 003520      QIO$S   #IO.ATA,#LUN.TT,....,<#AST2>
698 003562      RETURN
699      ;
700      ;
701      ;
702      ;
703      ;
704      ;
705      ;
706 003564      KILL:
707 003564 042767 000004 174242      BIC    #LOOP,BASE      ;CLEAR LOOP FLAG
708 003572 032767 000200 174234      BIT    #ASTFLG,BASE      ;HAS A QIO BEEN ISSUED?
709 003600 001423      BEQ    1$          ;NO, DO NOT DETACH
710 003602 042767 000200 174224      BIC    #ASTFLG,BASE      ;CLEAR QIO FLAG
711 003610      QIO$S   #IO.DET,#LUN.TT
712 003650      1$: RETURN

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

714      ;
715      ;      AST·FOR·LOOP·ON·TEST·Q10·
716      ;
717 003652·      ;      AST:
718 003652· 012667 174310      MOV·      (SP)+,ASTWRD·      ;GET·CHAR·OFF·STACK·
719 003656 042767 000004 174150      BIC·      #LOOP,BASE·      ;CLEAR·LOOP·FLAG·
720 003664      ;
721      ;
722      ;      AST·FOR·BREAKPOINT·Q10·
723      ;
724 003672·      ;      AST2:
725 003672· 012667 174270      MOV·      (SP)+,ASTWRD·
726 003676      SETF#S· #EFN,4      ;SET·EVENT·FLAG·FRO·TERMINAL·INTERRUPT·
727      ;
728 003710      ;      ASTX$S·

```

```

730      ;      SCAN A TABLE FOR A VALID COMMAND/MNEMONIC.
731      ;
732      ;      INPUT:
733      ;      R0 = NUMBER OF ENTRIES IN COMMAND TABLE.
734      ;      R1 -> CHAR STRING IN GCML COMMAND LINE.
735      ;      R2 -> TOP OF COMMAND TABLE.
736      ;
737      ;      OUTPUT:
738      ;      R1 -> ROUTINE THAT GOVERNS THE COMMAND (IF MATCH WAS MADE)
739      ;      R1 -> CHAR STRING IN COMMAND LINE (IF NO MATCH WAS MADE)
740      ;      R0 = RELATIVE POSITION OF MATCHED ENTRY IN TABLE.
741      ;
742      003716      SCAN::
743      003716      010346      MOV.      R3, -(SP)      ;SAVE R3
744      003720      010046      MOV.      R0, -(SP)      ;SAVE # ENTRIES
745      003722      010146      MOV.      R1, -(SP)      ;SAVE POINTER TO BEGINNING OF STRING
746      ;
747      003724      011601      FNOUT1: MOV.      (SP), R1      ;POINT TO NON-BLANK IN COMMAND LINE
748      003726      012703      000002      MOV.      #2, R3      ;NUMBER OF CHARS IN NON-BLANK FIELD
749      003732      122122      FNIN1:  CMPB.      (R1)+, (R2)+      ;DOES COMMAND LINE MATCH TABLE ENTRY
750      003734      001003      BNE.      FNOUT2      ;NO, TRY NEXT TABLE ENTRY
751      003736      005303      DEC.      R3      ;SUB FROM LOOP COUNT
752      003740      001374      BNE.      FNIN1
753      003742      000411      BR.      FNMTCH      ;COMMAND FOUND IN TABLE
754      003744      060302      FNOUT2: ADD.      R3, R2      ;ADD # UNCOMPARED CHARS TO POINTER
755      003746      005202      INC.      R2      ;THEN ADJUST TO NEXT TABLE ENTRY
756      003750      005300      DEC.      R0      ;SUB FROM OUTER LOOP COUNT
757      003752      001364      BNE.      FNOUT1      ;TRY AGAIN
758      003754      012601      MOV.      (SP)+, R1      ;RESTORE R1
759      003756      012600      MOV.      (SP)+, R0
760      003760      012603      MOV.      (SP)+, R3
761      003762      000261      SEC.
762      003764      RETURN.      ;COMMAND NOT IN TABLE
763      ;
764      003766      010201      FNMTCH: MOV.      R2, R1      ;POINT R1 AT RTN ADDR IN TABLE
765      003770      062706      000002      ADD.      #2, SP      ;POINT TO INCOMING R0 ON STACK
766      003774      012602      MOV.      (SP)+, R2      ;GET TOTAL # TABLE ENTRIES
767      003776      160002      SUB.      R0, R2      ;GET POSITION OF MATCHED ENTRY
768      004000      010200      MOV.      R2, R0      ;PUT IN R0 FOR RETURN
769      004002      012603      MOV.      (SP)+, R3      ;RESTORE R3
770      004004      000241      CLC.
771      004006      RETURN.

```



```

773      ;      FIND THE NEXT NON-BLANK IN THE COMMAND BUFFER,
774      ;      THEN FIND THE LENGTH OF THE STRING THAT STARTS WITH THAT CHARACTER.
775      ;
776      ;      INPUT:
777      ;      GCMLN - NUMBER OF UNPROCESSED BYTES IN COMMAND LINE.
778      ;      GCMPT - ADDR OF NEXT UNPROCESSED POSITION IN COMMAND LINE.
779      ;
780      ;      OUTPUT:
781      ;      R1 -> STRING. R0 - LENGTH OF STRING.
782      ;      GCMLN, GCMPT UPDATED FOR NEXT ENTRY INTO THIS ROUTINE.
783      ;
784      ;      THIS ROUTINE IS DESIGNED TO BE ENTERED A NUMBER OF TIMES
785      ;      IN THE PARSING OF A COMMAND LINE. THE FIELDS GCMLN AND
786      ;      GCMPT ARE REFRESHED WHEN A NEW COMMAND LINE IS READ
787      ;      (SEE THE SUBROUTINE 'GCONLY').
788      ;
789      ;
790 004010      FIND::
791 004010      MOV     R2, -(SP)          ;SAVE R2.
792 004012      MOV     GCMLN, R1        ;#. BYTES REMAINING IN COMMAND BUFFER.
793 004016      BEQ     FSECX           ;THERE ARE NONE.
794 004020      MOV     GCMPT, R2        ;LOAD CURRENT POINTER.
795 004024      1$:      CMPB    #40, (R2)    ;LOOK FOR A BLANK.
796 004030      BEQ     10$            ;OK. BUMP TO NEXT CHAR.
797 004032      CMPB    #'', (R2)       ;COMMA IN COMMAND LINE.
798 004036      BNE     2$            ;TREAT COMMA AS BLANK.
799 004040      10$:     INC     R2        ;BUMP POINTER.
800 004042      DEC     R1             ;SUB FROM REMAINING LENGTH.
801 004044      BNE     1$            ;
802 004046      BR      FSECX         ;NO NON-BLANK FOUND.
803      ;
804 004050      2$:      MOV     R2, -(SP)    ;TEMP SAVE POINTER TO BEGINNING OF STRING.
805 004052      CLR     R0             ;CLEAR CHAR COUNT.
806 004054      3$:      CMPB    #40, (R2)    ;LOOK FOR A BLANK.
807 004060      BEQ     4$            ;FOUND END OF STRING.
808 004062      CMPB    #'', (R2)       ;TREAT COMMAS AS BLANKS.
809 004066      BEQ     4$            ;
810 004070      INC     R2             ;BUMP POINTER.
811 004072      INC     R0             ;BUMP CHAR COUNT.
812 004074      DEC     R1             ;SUB FROM BYTES REMAINING.
813 004076      BNE     3$            ;
814      ;
815 004100      4$:      MOV     R2, GCMPT    ;SAVE POINTER FOR NEXT TIME.
816 004104      MOV     R1, GCMLN         ;SAVE BYTES REMAINING FOR NEXT TIME.
817 004110      MOV     (SP)+, R1        ;POINTER TO BEGINNING OF STRING.
818 004112      MOV     (SP)+, R2        ;RESTORE R2.
819 004114      CLC
820 004116      RETURN
821      ;
822 004120      FSECX:  MOV     (SP)+, R2    ;RESTORE R2.
823 004122      SEC
824 004124      RETURN

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

883 004276				GALL	\$DIV		:REDUCE FACTOR
884 004302	005302			DEC	R2		:SUB FROM LOOP COUNT
885 004304	001352			BNE	HLOOP		
886 004306	000403			BR	PCLCX		:EXIT
887				:			
888 004310	012766	177777	000014	PSECK:	MOV	*+1,12,(SP)	:SET COND CODE INDICATOR
889 004316				PCLCX:	RESTOR	R0,R1,R2,R3,R4,R5	
890				:			
891 004332	005726			TST	(SP)+		:GET COND CODE INDICATOR
892 004334	002402			BLT	1\$:DO SEC
893 004336	000241			CLC			
894 004340	000401			BR	PACKX		:AND RETURN
895 004342	000261			1\$:	SEC		
896 004344				PACKX:	RETURN		

```

898      ;
899      ;      CONVERT A VALUE FROM BINARY TO PRINTABLE FORM.
900      ;      R1 = WORD TO BE CONVERTED.
901      ;      R5 -> PRINT LINE.
902      ;
903      ;
904 004345      UNPK::
905 004345      ;      SAVE R0,R1,R2,R3,R4
906      ;
907 004360 062705 000004      ADD #4,R5      ;DO LAST CHAR FIRST.
908 004364 012702 000004      MOV #4,R2      ;NUMBER OF HEX DIGITS FOR A WORD.
909 004370 010100      MOV R1,R0      ;SUBRTH EXPECTS DIVIDEND IN R0
910 004372 012701 000020      1$: MOV #16,R1      ;LOAD DIVIDOR.
911 004376      CALL $DIV
912 004402 012703 000475      MOV #TRTBL2,R3      ;POINT TO TRANSLATE TABLE.
913 004406 060103      ADD R1,R3      ;ADD 4 BIT VALUE.
914 004410 111345      MOVB (R3),-(R5)      ;MOVE CHAR TO PRINT LINE.
915 004412 005302      DEC R2      ;DEC INNER LOOP COUNT.
916 004414 001366      BNE 1$
917 004416 062705 000005      ADD #5,R5      ;BUMP PRINT LINE POINTER.
918      ;
919 004422      RESTOR R0,R1,R2,R3,R4
920 004434      RETURN

```

```

922.      ;
923.      ;
924.      ;      WRITE A PRINT LINE TO TT0
925.      ;
926.      ;
927. 004436 CONSOL:
928. 004436      SAVE      R0,R1
929.      ;
930. 004442      012700      000120      MOV      #00,R0      ;PRINT BUFFER BYTE COUNT
931. 004446      012701      000666      MOV      #PRINT+78,R1      ;POINT PAST END OF BUFFER
932. 004452      122741      000040      1$: CMPB      #40,-(R1)      ;LOOK FOR A NON-BLANK
933. 004456      001003      BNE      2$      ;OK, WRITE LINE
934. 004460      005300      DEC      R0      ;DEC CHAR COUNT
935. 004462      001373      BNE      1$
936. 004464      000440      BR      ABEND2      ;NO NON-BLANKS?
937.      ;
938. 004466      2$: QIOW$S      #IO.WVB,#LUN.TT,#EFN.1,,#STAT,,<#PRINT-2,R0>,ABEND2
939.      ;
940.      ;
941. 004544      012701      000550      MOV      #PRINT,R1      ;POINT TO STRING
942. 004550      112721      000040      4$: MOVB      #40,(R1)+      ;CLEAR LINE TO BLANKS
943. 004554      005300      DEC      R0      ;DEC LOOP COUNT
944. 004556      001374      BNE      4$
945.      ;
946. 004560      RESTOR      R0,R1
947. 004564      RETURN
948.      ;
949. 004566      ABEND2: ABRT$S      #MYSELF

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

ABEND	005064R	BYTE22	= 000026	BYTE74	= 000112	DISPGS	= 100000	F.EFBK	= 000010
ABEND2	004566R	BYTE23	= 000027	BYTE75	= 000113	DMAAWR	= 000005	F.EFN	= 000050
ALUCKE	= 040000	BYTE24	= 000030	BYTE76	= 000114	DMARRD	= 000003	F.EOBB	= 000032
ALUOE	= 004000	BYTE25	= 000031	BYTE77	= 000115	DMARWR	= 000004	F.ERR	= 000052
APLACE	000036RG	BYTE26	= 000032	BYTE78	= 000116	EFBUF	= 000014RG	F.FACC	= 000043
ASC12	001665RG	BYTE27	= 000033	BYTE79	= 000117	EFN.1	= 000001 G	F.FBYY	= 000014
AST	003652R	BYTE28	= 000034	BYTE80	= 000120	EFN.2	= 000002 G	F.FNAM	= 000110
ASTFLG	= 000200 G	BYTE29	= 000035	BYTE81	= 000121	EFN.3	= 000003 G	F.FNB	= 000102
ASTWRD	000166RG	BYTE30	= 000036	BYTE82	= 000122	EFN.4	= 000004 G	F.FTYP	= 000116
AST2	003672R	BYTE31	= 000037	BYTE83	= 000123	ENBR	= 010000	F.FVER	= 000120
A01	= 010000	BYTE32	= 000040	BYTE84	= 000124	ENDMEM	= 004634RG	F.HIBK	= 000004
BASE	000034RG	BYTE33	= 000041	BYTE85	= 000125	ENDTST	= 004620RG	F.LUN	= 000042
BCE	= ***** GX	BYTE34	= 000042	BYTE86	= 000126	ENFILE	= 004630RG	F.MBC1	= 000055
BINWD	000032RG	BYTE35	= 000043	BYTE87	= 000127	ERR1	= 004724RG	F.MBFG	= 000056
BITVAL	= 000000	BYTE36	= 000044	BYTE88	= 000130	ERR10	= 004660RG	F.NRBD	= 000024
BIT0	= 000001	BYTE37	= 000045	BYTE89	= 000131	ERR11	= 004654RG	F.NREC	= 000030
BIT1	= 000002	BYTE38	= 000046	BYTE90	= 000132	ERR12	= 004650RG	F.OVBS	= 000030
BIT10	= 002000	BYTE39	= 000047	BYTE91	= 000133	ERR2	= 004720RG	F.RACC	= 000016
BIT11	= 004000	BYTE40	= 000050	BYTE92	= 000134	ERR3	= 004714RG	F.RATT	= 000001
BIT12	= 010000	BYTE41	= 000051	BYTE93	= 000135	ERR4	= 004710RG	F.RCNM	= 000034
BIT13	= 020000	BYTE42	= 000052	BYTE94	= 000136	ERR5	= 004704RG	F.RCTL	= 000017
BIT14	= 040000	BYTE43	= 000053	BYTE95	= 000137	ERR6	= 004700RG	F.RSIZ	= 000002
BIT15	= 100000	BYTE44	= 000054	BYTE96	= 000140	ERR7	= 004674RG	F.RTYP	= 000000
BIT2	= 000004	BYTE45	= 000055	BYTE97	= 000141	ERR8	= 004670RG	F.SEGN	= 000100
BIT3	= 000010	BYTE46	= 000056	BYTE98	= 000142	ERR9	= 004664RG	F.SPDV	= 000072
BIT4	= 000020	BYTE47	= 000057	BYTE99	= 000143	ERWORD	= 000030RG	F.SPUN	= 000074
BIT5	= 000040	BYTE48	= 000060	BYTVAL	= 000144	EXIT	= 002370R	F.STBK	= 000036
BIT6	= 000100	BYTE49	= 000061	CBKALL	= 001000	FAHIGH	= 000242RG	F.UNIT	= 000136
BIT7	= 000200	BYTE50	= 000065	CBKCLK	= 000400	FALOW	= 000244RG	F.URBD	= 000020
BIT8	= 000400	BYTE51	= 000066	CDHIGH	= 000232RG	FD.CCL	= ***** GX	F.VBN	= 000064
BIT9	= 001000	BYTE52	= 000064	CDLOW	= 000234RG	FD.REC	= ***** GX	F.VBSZ	= 000060
BLHIGH	000256PG	BYTE53	= 000065	CL	= 002450RG	FD.TTY	= ***** GX	GCMBLK	= 001666RG
BLLW	000260RG	BYTE54	= 000066	CMILUN	= 000002	FIND	= 004010RG	GCMBUF	= 000040RG
BPTISR	003330R	BYTE55	= 000067	CNOBRE	= 100000	FNIN1	= 003732R	GCMLEN	= 000162RG
BREAK	= 004000 G	BYTE56	= 000070	COMMX	= 002334RG	FNMTCH	= 003766R	GCMPLN	= 000164RG
BUFCX2	002750R	BYTE57	= 000071	CONSOL	= 004436RG	FNOUT1	= 003724R	GCONLY	= 003254RG
BUFSET	002612RG	BYTE58	= 000072	CP	= ***** GX	FNOUT2	= 003744R	GE.BIF	= 177775
BUFSM	002630RG	BYTE59	= 000073	CPCCEN	= 010000	FNUM	= 000006	GE.CLO	= 000004
BUFS2	002574RG	BYTE60	= 000074	CPREAD	= 040000	FSECX	= 004120R	GE.COM	= 000001
BUFS3	002556RG	BYTE61	= 000075	CPURTE	= 020000	FTBL	= 000516R	GE.CON	= 000020
BUFS4	002540RG	BYTE62	= 000076	CSADDR	= 000004	F.ACTL	= 000076	GE.EOF	= 177766
BUFX2	002762R	BYTE63	= 000077	CSEQCI	= 100000	F.ALOC	= 000040	GE.IND	= 000002
BYTE0	= 000000	BYTE64	= 000100	CSHIGH	= 000226RG	F.BBFS	= 000062	GE.IOR	= 177777
BYTE1	= 000001	BYTE65	= 000101	CSLOW	= 000230RG	F.BDB	= 000070	GE.LC	= 000010
BYTE10	= 000012	BYTE66	= 000102	CSOE	= 000040	F.BGBC	= 000057	GE.MDE	= 177774
BYTE11	= 000013	BYTE67	= 000103	CSP1	= 002410RG	F.BKDN	= 000026	GE.OPR	= 177776
BYTE12	= 000014	BYTE68	= 000104	CSURTE	= 000100	F.BKDS	= 000020	GE.RBG	= 177730
BYTE13	= 000015	BYTE69	= 000105	DATA1	= 000206RG	F.BKEF	= 000050	GE.SIZ	= 000040
BYTE14	= 000016	BYTE70	= 000106	DATA2	= 000210RG	F.BKP1	= 000051	G.CHLD	= 000146
BYTE15	= 000017	BYTE71	= 000107	DATA3	= 000212RG	F.BKST	= 000024	G.DPRM	= 000160 G
BYTE16	= 000020	BYTE72	= 000110	DATA4	= 000214RG	F.BKVB	= 000064	G.ERR	= 000140
BYTE17	= 000021	BYTE73	= 000111	DBR.RD	= 000001	F.CHR	= 000075	G.ISIZ	= 000020
BYTE18	= 000022			DB#CPP	= 001457	F.CNTG	= 000034	G.MDE	= 000060
BYTE19	= 000023			DB#SPT	= 000026	F.DFNB	= 000046	G.MODE	= 000144
BYTE2	= 000002			DB#TPC	= 000023	F.DSPT	= 000044	G.RSDS	= 000142
BYTE20	= 000024					F.DVNM	= 000134	G.SIZE	= 000224
BYTE21	= 000025								

HANG. 003426RG.	OLDVEC 000006RG	Q\$LBP. = 000001	S\$S1 = 000010	T\$RSET= 040000
HANG2. 003512RG.	ONCE. = 000100 G	Q\$LCD= 000003	S\$S2. = 000014	T\$SC = 000022
HLHIGH. 000252RG.	OUT. = 002000 G	Q\$LDMD= 000004	S\$BFHD= 000020	T\$CLK= 020000
HLLOW. 000254RG.	OUT1 004644RG	Q\$LDPP= 002000	S\$FATT= 000016	T\$SEG1= 000000
HLOOP. 004232R.	PACK. 004126RG	Q\$LHP. = 010000	S\$FDB= 000140	T\$SEG2= 000001
INCVL. 000200RG.	PACKX. 004344R.	Q\$MNC. = 140000	S\$FNAM= 000006	T\$SEG3= 000002
IO.ATA= ***** G.	PAR\$\$\$= 000027	Q\$MR. = 000052	S\$FNB= 000036	T\$SO = 000001
IO.DET= ***** G.	PCLCX. 004316R.	Q\$MRP. = 000040	S\$FNBW= 000017	T\$UBUS= 100000
IO.RVB= ***** G.	PDATA. 002764RG	Q\$MRP2= 000240	S\$FNTY= 000004	T\$ICLK= 000400
IO.WVB= ***** G.	PDCX. 003150R.	Q\$MSC. = 040000	S\$FTYP= 000002	T\$BBEN= 000020
KILL 003564RG	PDCX. 003142R.	Q\$MSET= 000004	S\$NFEN= 000020	UBD. IN= 000020
LHHIGH. 000246RG.	PDDX. 003154R.	Q\$MSP. = 100000	TD\$CTR= 176370	UNPK 004346RG.
LHLOW. 000250RG.	PDVX. 003134R.	Q\$NCLK= 176000	TD\$CTW= 176360	UPLIM. 000204RG.
LOADER. 000010R.	PLB. = 000010	Q\$PP. = 000100	TD\$INL= 004000	WORD0 = 000000
LOC.EN= 000100	PLC. = 000020	Q\$PPSW= 000320	TD\$MEM= 000270	WORD1 = 000002
LOC.WA= 040000	PLD. = 000030	Q\$PP2. = 000300	TD\$OAR= 176344	WORD10= 000024
LOC.WB= 100000	PLRW. = 000200	Q\$QHLT= 000013	TD\$OTR= 176346	WORD11= 000026
LOOP. = 000004 G.	PLR.EN= 000200	Q\$QL. = 000043	TD\$ORD= 000274	WORD12= 000030
LOOPR. 003404RG.	PPCR. = ***** GX.	Q\$QLA. = 000053	TD\$SW. = 176376	WORD13= 000032
LOOPX. 003510R.	PPS. = ***** GX.	Q\$QLB. = 000054	TD\$TAR= 176372	WORD14= 000034
LUN.TT= 000001 G.	PRDATA 003156RG	Q\$QLR. = 000001	TD\$TAW= 176362	WORD15= 000036
MAREN1= 000001	PRDX. 003252R.	Q\$QW. = 000042	TD\$TDR= 176374	WORD16= 000040
MAREN2= 004000	PRINT. 000550RG	Q\$RDCD= 000005	TD\$TDW= 176364	WORD17= 000042
MARLOD= 010000	PSECK. 004310R.	Q\$RDMD= 000006	TIME. 004624RG.	WORD18= 000044
MAROUT= 000002	QR\$CR1= 176420	Q\$REBK= 001000	TRTBL. 000276RG.	WORD19= 000046
MAR.LO= 002000	QR\$CR2= 176422	Q\$RNC. = 006000	TRTBL2. 000475RG.	WORD2= 000004
MAR.OU= 000040	QR\$LBR= 176424	Q\$RSC. = 004000	TSKTCB. 000004RG.	WORD20= 000050
MBKALL= 001000	QXHIGH 000236RG	Q\$RSET= 000010	T\$AD. = 000020	WORD21= 000052
MBKCLK= 000400	QXLOW. 000240RG	Q\$SM. = 100000	T\$BA. = 000002	WORD22= 000054
MDHIGH. 000222RG.	Q\$ATTN= 000100	Q\$SP. = 000120	T\$BD. = 000010	WORD23= 000056
MDLOW. 000224RG.	Q\$BCL. = 000001	Q\$SP2. = 000340	T\$BSO. = 100000	WORD24= 000060
MEND. 000176RG.	Q\$CCCP= 000040	RESEL. 002272R.	T\$BT. = 000020	WORD25= 000062
MHADR0= 000100	Q\$CHB. = 000400	RGQ.EN= 000200	T\$BTAR= 000030	WORD26= 000064
MHIGH. 000216RG.	Q\$CHRL= 000200	RGQ.VA= 020000	T\$BD. = 002000	WORD27= 000066
MLEFT= 000002	Q\$CLR. = 000040	RP. = 001000 G.	T\$CD. = 000100	WORD28= 000070
MLOW. 000220RG.	Q\$CNC. = 030000	RSPCNT. 000202RG.	T\$CLK. = 002000	WORD29= 000072
MMD. = 000004	Q\$CP. = 000060	RTNPT. 000170RG.	T\$DISK= 000200	WORD3 = 000006
MMDRTE= 000010	Q\$CPCC= 000010	SCAN. 003716RG.	T\$DRD. = 000004	WORD30= 000074
MMDRE= 100000	Q\$CP2. = 000260	SDHIGH. 000272RG.	T\$MEM= 010000	WORD31= 000076
MREN1 = 000001	Q\$CSC. = 010000	SDLOW. 000274RG.	T\$FSAA= 000000	WORD32= 000100
MREN2 = 020000	Q\$CSEL= 000360	SEQ.CI= 000010	T\$FSAB= 000004	WORD33= 000102
MRP. = ***** GX.	Q\$CSET= 000002	SET. 002644R.	T\$FSAC= 000014	WORD34= 000104
MSTR1. 000172RG.	Q\$CSP. = 020000	SPCR. = ***** GX.	T\$FSB2= 000010	WORD35= 000106
MSTR2. 000174RG.	Q\$DMA. = 000001	SPS. = ***** GX.	T\$IB. = 000026	WORD36= 000110
MSYN. = 000040	Q\$ENBK= 040000	SQHIG. 000262RG.	T\$IBAR= 000024	WORD37= 000112
MYSELF. 000000RG.	Q\$ENOP= 020000	SQLOW. 000264RG.	T\$IBE. = 020000	WORD38= 000114
N. = 000144	Q\$FAL. = 004000	SRHIGH. 000266RG.	T\$IBF. = 040000	WORD39= 000116
NEQLB. = 010000 G.	Q\$FC. = 000045	SRLOW. 000270RG.	T\$ICD. = 000040	WORD4 = 000010
N.DID. = 000024	Q\$FO. = 000044	START. 002174R.	T\$MODE= 004000	WORD40= 000120
N.DVNM= 000032	Q\$FP. = 000046	STAT. 000024RG.	T\$OB. = 000036	WORD41= 000122
N.FID. = 000000	Q\$HBF. = 000002	STOP. 004640RG.	T\$OBE. = 004000	WORD42= 000124
N.FNAM= 000006	Q\$ICP. = 000006	S\$CLR. = 000000	T\$OBF. = 010000	WORD43= 000126
N.FTYP= 000014	Q\$IHB. = 000003	S\$LA. = 000001	T\$OBRA= 000034	WORD44= 000130
N.FVER= 000016	Q\$IHRL= 000002	S\$OB. = 000005	T\$OBWA= 000032	WORD45= 000132
N.NEXT= 000022	Q\$IMRP= 000007	S\$OR. = 000006	T\$OUTA= 100000	WORD46= 000134
N.STAT= 000020	Q\$LBD. = 001000	S\$QX. = 000004	T\$RDO. = 000200	WORD47= 000136
N.UNIT= 000034	Q\$LBDP= 001001	S\$SP. = 000007	T\$RNR. = 000040	WORD48= 000140

QMAIN: MACRO-11:110 27-MAR-80 15:17 PAGE 24-3
SYMBOL TABLE:

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

WORD49 = 000142	WORD62 = 000174	WORD76 = 000230	WORD9 = 000022	\$CEFI = ***** GX
WORD5 = 000012	WORD63 = 000176	WORD77 = 000232	WORD90 = 000264	\$DIV = ***** GX
WORD50 = 000144	WORD64 = 000200	WORD78 = 000234	WORD91 = 000266	\$DRDSE = ***** GX
WORD51 = 000146	WORD65 = 000202	WORD79 = 000236	WORD92 = 000270	\$MUL = ***** GX
WORD52 = 000150	WORD66 = 000204	WORD8 = 000020	WORD93 = 000272	\$TKTCB = ***** GX
WORD53 = 000152	WORD67 = 000206	WORD80 = 000240	WORD94 = 000274	\$\$\$ = 002046R
WORD54 = 000154	WORD68 = 000210	WORD81 = 000242	WORD95 = 000276	\$\$\$ARG = 000002
WORD55 = 000156	WORD69 = 000212	WORD82 = 000244	WORD96 = 000300	\$\$\$T1 = 000067
WORD56 = 000160	WORD7 = 000016	WORD83 = 000246	WORD97 = 000302	\$\$\$T2 = 000027
WORD57 = 000162	WORD70 = 000214	WORD84 = 000250	WORD98 = 000304	.FSRCB = ***** G
WORD58 = 000164	WORD71 = 000216	WORD85 = 000252	WORD99 = 000306	.GCML1 = ***** G
WORD59 = 000166	WORD72 = 000220	WORD86 = 000254	WRDVAL = 000310	...PC1 = 001666R
WORD6 = 000014	WORD73 = 000222	WORD87 = 000256	WORDS = 000206R	...PC2 = 002070R
WORD60 = 000170	WORD74 = 000224	WORD88 = 000260	XTREAD = 001000	...PC3 = 001666R
WORD61 = 000172	WORD75 = 000226	WORD89 = 000262	XTWRITE = 000400	...TPC = 000020

. ABS. 000000 000
005116 001
\$\$FSR1 001020 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 7917 WORDS (31 PAGES)
DYNAMIC MEMORY: 9140 WORDS (35 PAGES)
ELAPSED TIME: 00:01:36
QMAIN,QMAIN/SP=C20,1JIM,C20,1JQMAIN

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

1      .TITLE CPSUB
2      .PSECT CPSUB
3
4      ;
5      ;   HARDWARE QUERY RESOLVER: "MANUAL" DEBUGGING AIDS.
6      ;   ***** PROTOTYPE VERSION *****
7
8      ;
9      ;   CP SUBROUTINES.
10
11      ;   CONTROL STORE SEQUENCING (CONTROL PROCESSOR)
12      ;   JUMP TO START ADDRESS (VIA BRANCH REGISTER)
13
14      ;   INPUT:
15      ;   2(SP)  START ADDRESS.
16
17      ;   OUTPUT:
18      ;   CP SEQUENCER SET TO START ADDRESS.
19
20      ;
21      ;   CPSUB:
22      ;   SEQCS:
23
24      ;   STOP CP AND MRP CLOCKS
25
26      ;
27      ;   CLR      -(SP)                ;CLEAR NOTHING.
28      ;   MOV      #0$NCLK, -(SP)      ;SET NO CLOCKS.
29      ;   CALL     CSR1
30
31      ;
32      ;   GO TO LOAD MODE (AND CLEAR EVERYTHING ELSE)
33
34      ;
35      ;   CLR      QR$CR2                ;DO IT
36
37      ;
38      ;   SELECT BRANCH REGISTER IN CP.
39
40      ;
41      ;   CLR      -(SP)                ;DEFINE BR REG SELECT.
42      ;   CALL     CPCRA                ;WRITE CR (WITHOUT AUTOMATIC BR INHIBIT)
43
44      ;
45      ;   CLOCK CP ONCE.
46
47      ;
48      ;   MOV      #0$CNC, -(SP)        ;CLEAR CP NO CLOCKS.
49      ;   MOV      #0$CSP, -(SP)      ;SET CP SINGLE CLOCK
50      ;   CALL     CSR1
51
52      ;
53      ;   MOVE INSTRUCTION AND ADDRESS TO BRANCH REGISTER VIA LOAD BUS.
54      ;   ISSUE CLOCK TO PUT START ADDRESS CONTENTS AT INPUTS TO PLR.
55
56      ;
57      ;   BIS      #BIT13, 2(SP)        ;OR JMP INSTRUCTION (02) INTO HIGH ORDER NIBBLE.
58      ;   MOV      2(SP), -(SP)        ;COPY ARGUMENT INTO STACK FOR LBCP CALL.
59      ;   CALL     LBCP                ;DO MOVE -- ISSUE CLOCK.
60
61      ;
62      ;   MOV      (SP), 2(SP)          ;MOVE RETURN ADDRESS DOWN STACK.
63      ;   TST      (SP)+                ;POINT TO RETURN ADDRESS.
64      ;   RETURN

```

```

53      ;
54      ; CONTROL STORE SEQUENCING (CONTROL PROCESSOR)
55      ; SET START ADDRESS FOR WRITE
56      ;
57      ; INPUT:
58      ; 2(SP) START ADDRESS
59      ;
60      ; OUTPUT:
61      ; CP SEQUENCER SET TO START ADDRESS
62      ;
63 000066      WRTCS:
64      ;
65      ; STOP CP AND MRP CLOCKS
66      ;
67 000066 005046      CLR      -(SP)          ;CLEAR NOTHING
68 000070 012746 176000      MOV      #Q$NCLK,-(SP)      ;SET NO CLOCKS
69 000074      CALL      CSR1
70      ;
71      ; GO TO LOAD MODE (AND CLEAR EVERYTHING ELSE)
72      ;
73 000100 005067 176422      CLR      QR$CR2          ;DO IT
74      ;
75      ; SELECT BRANCH REGISTER IN CP
76      ;
77 000104 005046      CLR      -(SP)          ;DEFINE BR REG SELECT
78 000106      CALL      CPCRA          ;WRITE CR (WITHOUT AUTOMATIC BR INHIBIT)
79      ;
80      ; CLOCK CP ONCE
81      ;
82 000112 012746 030000      MOV      #Q$CNC,-(SP)      ;CLEAR CP NO CLOCKS
83 000116 012746 020000      MOV      #Q$CSP,-(SP)      ;SET CP SINGLE CLOCK
84 000122      CALL      CSR1
85      ;
86      ; MOVE INSTRUCTION AND ADDRESS TO BRANCH REGISTER VIA LOAD BUS
87      ;
88 000126 052766 020000 000002      BIS      #BIT13,2(SP)      ;OR JMP INSTRUCTION (02) INTO HIGH ORDER NIBBLE
89 000134 016646 000002      MOV      2(SP),-(SP)      ;COPY ARGUMENT INTO STACK FOR LBCP CALL
90 000140      CALL      LBCP          ;DO MOVE - FIRST CLOCK
91      ;
92 000144      RETURN

```

```
94      ;
95      ;
96      ; DATA TRANSFER
97      ; LOD BUS REGISTER TO A DESTINATION ON THE CP BUS
98      ;
99      ; INPUT:
100     2(SP) DATA FOR PRE-SELECTED CP DESTINATION
101     000146
102     000146 016667 000002 176424 LBCP::
103     000154 012746 001001
104     000160 052716 000360
105     000164 012746 176000
106     000170 052716 000260
107     000174
108     ;
109     000200 012746 030000
110     000204 012746 021000
111     000210
112     ;
113     ;
114     ; DE-SELECTION
115     000214 012746 001001
116     000220 052716 000360
117     000224 012746 176000
118     000230
119     ;
120     000234 011666 000002
121     000240 005726
122     000242
```

MOV. 2(SP),QR\$LBR ; MOVE DATA TO LOD BUS REG.
MOV. *(<Q\$LBD+Q\$LBP>),-(SP) ; CLR DRIVE AND PULSE
BIS. *(<Q\$CSEL>),(SP) ; CLR SELECTION BITS
MOV. *(<Q\$NCLK>),-(SP) ; SET NO-CLOCKS
BIS. *Q\$CP2,(SP) ; SELECT CP
CALL CSR1

MOV. *Q\$CNC,-(SP) ; CLEAR CP NO-CLOCK BITS
MOV. *(<Q\$CSP+Q\$LBD>),-(SP) ; SET CP CLOCK
CALL CSR1

MOV. *(<Q\$LBD+Q\$LBP>),-(SP) ; CLEAR DRIVE AND PULSE
BIS. *(<Q\$CSEL>),(SP) ; CLR SELECTION BITS
MOV. *(<Q\$NCLK>),-(SP) ; SET NO-CLOCKS
CALL CSR1

MOV. (SP),2(SP) ; MOVE RETURN ADDRESS DOWN STACK
TST. (SP)+ ; POINT TO RETURN ADDRESS
RETURN.

```
124      ;
125      ; DATA TRANSFER
126      ; LOD-BUS REGISTER TO A DESTINATION ON THE CP-BUS
127      ; SINGLE-CLOCK SEQUENCER ONLY
128      ;
129      ; INPUT:
130      ; 2(3P) DATA FOR PRE-SELECTED CP DESTINATION
131      ;
132      000244      LBCSC::
133      000244      016667      000002      176424      MOV      2(SP),QR$LBR      ;MOVE DATA TO LOD-BUS-REG
134      000252      012746      001001      MOV      *(<Q$LBD+Q$LBP>),-(SP)      ;CLR DRIVE AND PULSE
135      000256      052716      000360      BIS      *(<Q$CSEL>),(SP)      ;CLR SELECTION BITS
136      000262      012746      176000      MOV      *(<Q$NCLK>),-(SP)      ;SET NO-CLOCKS
137      000266      052716      000260      BIS      *Q$CP2,(SP)      ;SELECT CP
138      000272      CALL      CSR1
139      ;
140      000276      012746      030000      MOV      *Q$CNC,-(SP)      ;CLEAR CP NO-CLOCK BITS
141      000302      012746      011000      MOV      *(<Q$CSC+Q$LBD>),-(SP)      ;SET CP CLOCK
142      000306
143      ;
144      ; DE-SELECTION
145      ;
146      000312      012746      001001      MOV      *(<Q$LBD+Q$LBP>),-(SP)      ;CLR DRIVE AND PULSE
147      000316      052716      000360      BIS      *(<Q$CSEL>),(SP)      ;CLR SELECTION BITS
148      000322      012746      176000      MOV      *(<Q$NCLK>),-(SP)      ;SET NO-CLOCKS
149      000326      CALL      CSR1
150      ;
151      000332      011666      000002      MOV      (SP),2(SP)      ;MOVE RETURN ADDRESS DOWN STACK
152      000336      005726      TST      (SP)+      ;POINT TO RETURN ADDRESS
153      000340      RETURN
```

```
155      ;
156      ;      DATA TRANSFER TO LOD BUS REG FROM CP
157      ;
158      ;      OUTPUT:
159      ;      (SP) DATA FROM PRE-SELECTED CP SOURCE
160      ;
161 000342 CPLB::
162 000342 012746 001001 MOV. #<Q$LBD+Q$LBP>,-(SP) ;CLR DRIVE AND PULSE
163 000346 052716 000360 BIS. #<Q$CSEL>,(SP) ;CLR SELECTION BITS
164 000352 012746 176000 MOV. #<Q$NCLK>,-(SP) ;SET NO-CLOCKS
165 000356 052716 000260 BIS. #Q$CP2,(SP) ;SOURCE IS CP
166 000362 CALL CSR1
167 000366 011646 MOV. (SP),-(SP) ;MOVE RETURN ADDR UP STACK
168 000370 016766 176424 000002 MOV. QR$LBR,2(SP) ;MOVE DATA ONTO STACK
169      ;
170 000376 012746 000260 MOV. #Q$CP2,-(SP) ;CLEAR CP SELECT
171 000402 005046 CLR. -(SP) ;SET NOTHING
172 000404 CALL CSR1
173 000410 RETURN
```

```

175      ;
176      ;
177      ; CP CONTROL REGISTER LOADING
178      ;
179      ; INPUT:
180      ; 2(SP) BIT SETTING FOR CP CONTROL REGISTER
181      ;
182      000412      ;
183      000412 052766 100000 000002      ;
184      000420      ;
185      000420 016667 000002 176424      ;
186      000426 012746 001001      ;
187      000432 052716 000360      ;
188      000436 012746 000060      ;
189      000442      ;
190      ;
191      000446 005046      ;
192      000450 012746 000001      ;
193      000454      ;
194      ;
195      000460 012746 000061      ;
196      000464 005046      ;
197      000466      ;
198      ;
199      000472 011666 000002      ;
200      000476 005726      ;
201      000500      ;
202      ;
203      000001      ;

```

PCR:: BIS #CNOBREG,2(SP) ;AUTOMATIC INHIBIT OF BRANCH REGISTER
 CPCRA:: MOV 2(SP),QR#LBR ;CONTROL BITS DESTINED FOR CP
 MOV #<Q#LBD+Q#LBP>,-(SP) ;CLEAR DRIVE AND PULSE
 BIS #<Q#CSEL>,(SP) ;CLR SELECTION BITS
 MOV #Q#CP,-(SP) ;SELECT CP
 CALL CSRI
 CLR -(SP) ;CLEAR NOTHING
 MOV #Q#LBP,-(SP) ;SET PULSE
 CALL CSRI
 MOV #<Q#CP+Q#LBP>,-(SP) ;CLEAR CR SELECTION AND PULSE
 CLR -(SP) ;SET NOTHING
 CALL CSRI
 MOV (SP),2(SP) ;MOVE RETURN ADDRESS DOWN STACK
 TST (SP)+ ;POINT TO RETURN ADDRESS
 RETURN
 .END

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	PLR.EN = 000200	Q#RNC = 000000
ALUOE = 004000	BYTE43 = 000053	BYTE95 = 000137	QR#CR1 = 176420	Q#RSC = 004000
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	QR#CR2 = 176422	Q#RSET = 000010
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	QR#LBR = 176424	Q#SM = 100000
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	Q#ATTN = 000100	Q#SP = 000120
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	Q#BCL = 000001	Q#SP2 = 000340
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	Q#CCCP = 000040	RGQ.EN = 000200
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	Q#CHB = 000400	RGQ.VA = 020000
BIT12 = 010000	BYTE5 = 000005	CBKCLK = 000400	Q#CHRL = 000200	SEQCS = 000000RG 002
BIT13 = 020000	BYTE50 = 000062	CNOBRE = 100000	Q#CLR = 000040	SEQ.CI = 000010
BIT14 = 040000	BYTE51 = 000063	CPCCEN = 010000	Q#CNC = 030000	S#CLR = 000000
BIT15 = 100000	BYTE52 = 000064	CPCR = 000412RG	002.Q#CP = 000060	S#LA = 000001
BIT2 = 000004	BYTE53 = 000065	CPCRA = 000420RG	002.Q#CPCC = 000010	S#OB = 000005
BIT3 = 000010	BYTE54 = 000066	CPLB = 000342RG	002.Q#CP2 = 000260	S#OR = 000006
BIT4 = 000020	BYTE55 = 000067	CPREAD = 040000	Q#CSC = 010000	S#OX = 000004
BIT5 = 000040	BYTE56 = 000070	CPSUB = 000000R	002.Q#CSEL = 000360	S#SR = 000007
BIT6 = 000100	BYTE57 = 000071	CPWRITE = 020000	Q#CSET = 000002	S#S1 = 000010
BIT7 = 000200	BYTE58 = 000072	CSADDR = 000004	Q#CSP = 020000	S#S2 = 000014
BIT8 = 000400	BYTE59 = 000073	CSECCI = 100000	Q#DMA = 000001	TD#CTR = 176370
BIT9 = 001000	BYTE6 = 000006	CSDOE = 000040	Q#ENBK = 040000	TD#CTW = 176360
BYTE0 = 000000	BYTE60 = 000074	CSR1 = 000000 GX	Q#ENOP = 020000	TD#INL = 004000
BYTE1 = 000001	BYTE61 = 000075	CSWRITE = 000100	Q#FAL = 004000	TD#IEM = 000270
BYTE10 = 000012	BYTE62 = 000076	DBR.RD = 000001	Q#FC = 000045	TD#OAR = 176344
BYTE11 = 000013	BYTE63 = 000077	DB#CPP = 001457	Q#FO = 000044	TD#OTR = 176346
BYTE12 = 000014	BYTE64 = 000100	DB#SPT = 000026	Q#FP = 000046	TD#QRD = 000274
BYTE13 = 000015	BYTE65 = 000101	DB#TPC = 000023	Q#HBF = 000002	TD#SW = 176376
BYTE14 = 000016	BYTE66 = 000102	DISPGS = 100000	Q#ICP = 000006	TD#TAR = 176372
BYTE15 = 000017	BYTE67 = 000103	DMAWR = 000005	Q#IHB = 000003	TD#TAU = 176362
BYTE16 = 000020	BYTE68 = 000104	DMARRD = 000003	Q#IHL = 000002	TD#TDI = 176374
BYTE17 = 000021	BYTE69 = 000105	DMARWR = 000004	Q#IMRP = 000007	TD#TDW = 176364
BYTE18 = 000022	BYTE7 = 000007	ENBR = 010000	Q#LBD = 001000	T#AD = 000020
BYTE19 = 000023	BYTE70 = 000106	LBCP = 000146RG	002.Q#LBDP = 001001	T#BA = 000002
BYTE2 = 000002	BYTE71 = 000107	LBCSC = 000244RG	002.Q#LBP = 000001	T#BD = 000010
BYTE20 = 000024	BYTE72 = 000110	LOC.EN = 000100	Q#LDCD = 000003	T#BSO = 100000
BYTE21 = 000025	BYTE73 = 000111	LOC.WA = 040000	Q#LDMO = 000004	T#BT = 000020
BYTE22 = 000026	BYTE74 = 000112	LOC.WB = 100000	Q#LDPP = 002000	T#BTAR = 000030
BYTE23 = 000027	BYTE75 = 000113	MAREN1 = 000001	Q#LHP = 010000	T#BTD = 002000
BYTE24 = 000030	BYTE76 = 000114	MAREN2 = 004000	Q#MNC = 140000	T#CD = 000100
BYTE25 = 000031	BYTE77 = 000115	MARLOD = 010000	Q#MR = 000052	T#CLK = 002000
BYTE26 = 000032	BYTE78 = 000116	MAROUT = 000002	Q#MRP = 000040	T#DISK = 000200
BYTE27 = 000033	BYTE79 = 000117	MAR.LO = 002000	Q#MRP2 = 000240	T#DRD = 000004
BYTE28 = 000034	BYTE8 = 000010	MAR.OU = 000040	Q#MSC = 040000	T#EMEM = 010000
BYTE29 = 000035	BYTE80 = 000120	MBKALL = 001000	Q#MSET = 000004	T#FSAA = 000000
BYTE3 = 000003	BYTE81 = 000121	MBKCLK = 000400	Q#MSP = 100000	T#FSAB = 000004
BYTE30 = 000036	BYTE82 = 000122	MMADRD = 000100	Q#NCLK = 176000	T#FSAC = 000014
BYTE31 = 000037	BYTE83 = 000123	MMLEFT = 000002	Q#PP = 000100	T#FSB2 = 000010
BYTE32 = 000040	BYTE84 = 000124	MMOE = 000004	Q#PPSW = 000320	T#IB = 000026
BYTE33 = 000041	BYTE85 = 000125	MMWRTE = 000010	Q#PP2 = 000300	T#IBAR = 000024
BYTE34 = 000042	BYTE86 = 000126	MNOBRE = 100000	Q#QHLT = 000013	T#IBE = 020000
BYTE35 = 000043	BYTE87 = 000127	MREN1 = 000001	Q#QL = 000043	T#IBF = 040000
BYTE36 = 000044	BYTE88 = 000130	MREN2 = 020000	Q#QLA = 000053	T#ICD = 000040
BYTE37 = 000045	BYTE89 = 000131	MSYN = 000040	Q#QLB = 000054	T#MODE = 004000
BYTE38 = 000046	BYTE9 = 000011	N = 000144	Q#QLR = 000001	T#OB = 000036
BYTE39 = 000047	BYTE90 = 000132	PLB = 000010	Q#QW = 000042	T#OBF = 004000
BYTE4 = 000004	BYTE91 = 000133	PLC = 000020	Q#RDCD = 000005	T#OBF2 = 010000
BYTE40 = 000050	BYTE92 = 000134	PLD = 000030	Q#RDMD = 000006	T#OBRA = 000034
BYTE41 = 000051	BYTE93 = 000135	PLRW = 000200	Q#REBK = 001000	T#OBWA = 000032

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

CPSUB: M1110 27-MAR-80 14:45 PAGE: 10-2
SYMBOL: TA

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

T\$OUTA= 100000	WORD18= 000044	WORD4 = 000010	WORD61= 000172	WORD82= 000244
T\$RBD0= 000200	WORD19= 000046	WORD40= 000120	WORD62= 000174	WORD83= 000246
T\$RNB= 000040	WORD2= 000004	WORD41= 000122	WORD63= 000176	WORD84= 000250
T\$RSET= 040000	WORD20= 000050	WORD42= 000124	WORD64= 000200	WORD85= 000252
T\$SC= 000022	WORD21= 000052	WORD43= 000126	WORD65= 000202	WORD86= 000254
T\$SCLK= 020000	WORD22= 000054	WORD44= 000130	WORD66= 000204	WORD87= 000256
T\$SEG1= 000000	WORD23= 000056	WORD45= 000132	WORD67= 000206	WORD88= 000260
T\$SEG2= 000001	WORD24= 000060	WORD46= 000134	WORD68= 000210	WORD89= 000262
T\$SEG3= 000002	WORD25= 000062	WORD47= 000136	WORD69= 000212	WORD9= 000022
T\$SO= 000001	WORD26= 000064	WORD48= 000140	WORD7= 000016	WORD90= 000264
T\$UBUS= 100000	WORD27= 000066	WORD49= 000142	WORD70= 000214	WORD91= 000266
T\$1CLK= 000400	WORD28= 000070	WORD5= 000012	WORD71= 000216	WORD92= 000270
T\$0BEN= 000020	WORD29= 000072	WORD50= 000144	WORD72= 000220	WORD93= 000272
UBD: IN= 000020	WORD3= 000006	WORD51= 000146	WORD73= 000222	WORD94= 000274
WORD0= 000000	WORD30= 000074	WORD52= 000150	WORD74= 000224	WORD95= 000276
WORD1= 000002	WORD31= 000076	WORD53= 000152	WORD75= 000226	WORD96= 000300
WORD10= 000024	WORD32= 000100	WORD54= 000154	WORD76= 000230	WORD97= 000302
WORD11= 000026	WORD33= 000102	WORD55= 000156	WORD77= 000232	WORD98= 000304
WORD12= 000030	WORD34= 000104	WORD56= 000160	WORD78= 000234	WORD99= 000306
WORD13= 000032	WORD35= 000106	WORD57= 000162	WORD79= 000236	WORDVAL= 000310
WORD14= 000034	WORD36= 000110	WORD58= 000164	WORD8= 000020	WRTCS= 000066RG 002
WORD15= 000036	WORD37= 000112	WORD59= 000166	WORD80= 000240	XTREAD= 001000
WORD16= 000040	WORD38= 000114	WORD6= 000014	WORD81= 000242	XTWRITE= 000400
WORD17= 000042	WORD39= 000116	WORD60= 000170		

. ABS: 000000 000
000000 001
CPSUB: 000502 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3021 WORDS (12 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:43
CPSUB: CPSUB/SP=[20,1]IM,[20,1]CPSUB

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

1      .TITLE MRPSUB.
2 000000 .PSECT MRPSUB.
3
4      ;
5      ;   HARDWARE QUERY RESOLVER "MANUAL" DEBUGGING AIDS
6      ;   ***** PROTOTYPE VERSION *****
7
8      ;
9      ;   MRP SUBROUTINES
10
11      ;
12      ;   CONTROL STORE SEQUENCING (MATCH REPORT PROCESSOR)
13      ;   JUMP TO START ADDRESS (VIA BRANCH REGISTER)
14
15      ;
16      ;   INPUT:
17      ;   2(SP)  START ADDRESS
18
19      ;
20      ;   OUTPUT:
21      ;   MRP SEQUENCER SET TO START ADDRESS
22
23      ;
24      ;   MRPSUB:
25      ;   SEQM1:
26
27      ;
28      ;   STOP GP AND MRP CLOCKS
29
30      ;
31      ;   CLR      -(SP)                ; CLEAR NOTHING
32      ;   MOV      #Q$NCLK, -(SP)      ; SET NO CLOCKS
33      ;   CALL     CSR1
34
35      ;
36      ;   GO TO LOAD MODE (AND CLEAR EVERYTHING ELSE)
37
38      ;
39      ;   CLR      QR$CR2                ; DO IT
40
41      ;
42      ;   SELECT BRANCH REGISTER IN MRP
43
44      ;
45      ;   CLR      -(SP)                ; DEFINE BR REG SELECT
46      ;   CALL     MRP$CRA              ; WRITE CR (WITHOUT AUTOMATIC BR SUPPRESS)
47
48      ;
49      ;   CLOCK MRP ONCE
50
51      ;
52      ;   MOV      #Q$MNC, -(SP)        ; CLEAR MRP NO CLOCKS
53      ;   MOV      #Q$MSP, -(SP)        ; SET MRP SINGLE CLOCK
54      ;   CALL     CSR1
55
56      ;
57      ;   MOVE INSTRUCTION AND ADDRESS TO BRANCH REGISTER VIA LOAD BUS
58
59      ;
60      ;   BIS      #BIT13, 2(SP)        ; OR JMP INSTRUCTION (02) INTO HIGH ORDER NIBBLE
61      ;   MOV      2(SP), -(SP)        ; COPY ARGUMENT INTO STACK FOR LBMRP CALL
62      ;   CALL     LBMRP                ; DO MOVE - FIRST CLOCK
63
64      ;
65      ;   MOV      (SP), 2(SP)          ; MOVE RETURN ADDRESS DOWN STACK
66      ;   TST      (SP) +              ; POINT TO RETURN ADDRESS
67      ;   RETURN

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

52.      ;
53.      ;      CONTROL-STORE-SEQUENCING-(MATCH-REPORT-PROCESSOR)
54.      ;      SET-START-ADDRESS-FOR-WRITE.
55.      ;
56.      ;      INPUT:
57.      ;      Z(SP)      START-ADDRESS.
58.      ;
59.      ;      OUTPUT:
60.      ;      MRP-SEQUENCER-SET-TO-START-ADDRESS.
61.      ;
62. 000066      ;      WRTMM:
63.      ;
64.      ;      STOP-CP-AND-MRP-CLOCKS
65.      ;
66. 000066      005046      CLR.      -(SP)      ;CLEAR-NOTHING.
67. 000070      012746      176000      MOV.      #Q$NCLK,-(SP)      ;SET-NO-CLOCKS.
68. 000074      CALL.      CSRI
69.      ;
70.      ;      GO-TO-LOAD-MODE-(AND-CLEAR-EVERYTHING-ELSE)
71.      ;
72. 000100      005067      176422      CLR.      QR$CR2.      ;DO-IT
73.      ;
74.      ;      SELECT-BRANCH-REGISTER-IN-MRP.
75.      ;
76. 000104      005046      CLR.      -(SP)      ;DEFINE-BR-REG-SELECT.
77. 000106      CALL.      MRPCRA.      ;WRITE-CR-(WITHOUT-AUTOMATIC-BR-INHIBIT)
78.      ;
79.      ;      CLOCK-MRP-ONCE.
80.      ;
81. 000112      012746      140000      MOV.      #Q$MNC,-(SP)      ;CLEAR-MRP-NO-CLOCKS
82. 000116      012746      100000      MOV.      #Q$MSP,-(SP)      ;SET-MRP-SINGLE-CLOCK.
83. 000122      CALL.      CSRI
84.      ;
85.      ;      MOVE-INSTRUCTION-AND-ADDRESS-TO-BRANCH-REGISTER-VIA-LOAD-BUS.
86.      ;
87. 000126      052766      020000      000002      BIS.      #BIT13,2(SP)      ;OR-JMP-INSTRUCTION-(02) INTO-HIGH-ORDER-NIBBLE.
88. 000134      016646      000002      MOV.      2(SP),-(SP)      ;COPY-ARGUMENT-INTO-STACK-FOR-LBMRP-CALL
89. 000140      CALL.      LBMRP.      ;DO-MOVE--FIRST-CLOCK.
90.      ;
91. 000144      011666      000002      MOV.      (SP),2(SP)      ;MOVE-RETURN-ADDRESS-DOWN-STACK.
92. 000150      005726      TST.      (SP)+      ;POINT-TO-RETURN-ADDRESS.

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

94      ;
95      ;
96      ; DATA TRANSFER
97      ;
98      ; LOD BUS REGISTER TO A DESTINATION ON THE MRP BUS
99      ;
100     ; INPUT
101     ; 2(SP) DATA FOR PRE-SELECTED MRP DESTINATION
102     ;
103     ; LBMRP::
104     ;
105     ; MOV. 2(SP),QR$LBR. ;MOVE DATA TO LOD BUS REG.
106     ; MOV. *(Q$LBD+Q$LBP),-(SP) ;CLR DRIVE AND PULSE
107     ; BIS. *(Q$CSEL), (SP) ;CLR SELECTION BITS
108     ; MOV. *(Q$NCLK),-(SP) ;SET NO-CLOCKS
109     ; BIS. *(Q$MRP2), (SP) ;SELECT MRP
110     ; CALL. CSR1
111     ;
112     ; MOV. #Q$MNC, -(SP) ;CLEAR MRP NO-CLOCK BITS
113     ; MOV. *(Q$MSP+Q$LBD),-(SP) ;SET MRP CLOCK
114     ; CALL. CSR1
115     ;
116     ; DE-SELECTION
117     ;
118     ; MOV. *(Q$LBD+Q$LBP),-(SP) ;CLEAR DRIVE AND PULSE
119     ; BIS. *(Q$CSEL), (SP) ;CLR SELECTION BITS
120     ; MOV. *(Q$NCLK),-(SP) ;SET NO-CLOCKS
121     ; CALL. CSR1
122     ;
123     ; MOV. (SP),2(SP) ;MOVE RETURN ADDRESS DOWN STACK
124     ; TST. (SP)+ ;POINT TO RETURN ADDRESS
125     ;
126     ; RETURN

```

```

124      ;
125      ;      DATA TRANSFER
126      ;      LOD-BUS REGISTER TO A DESTINATION ON THE MRP-BUS
127      ;      SINGLE-CLOCK SEQUENCER ONLY
128      ;
129      ;      INPUT:
130      ;      2(SP) DATA FOR PRE-SELECTED MRP DESTINATION
131      ;
132      000250      LBMSC::
133      000250      016667      000002      176424      MOV.      2(SP),QR$LBR.      ;MOVE DATA TO LOD-BUS-REG.
134      000256      012746      001001      MOV.      #<Q$LBD+Q$LBP>,-(SP)      ;CLR DRIVE AND PULSE
135      000262      052716      000360      BIS.      #<Q$CSEL>,(SP)      ;CLR SELECTION BITS
136      000266      012746      176000      MOV.      #<Q$NCLK>,-(SP)      ;SET NO-CLOCKS
137      000272      052716      000240      BIS.      #Q$MRP2,(SP)      ;SELECT MRP
138      000276      CALL      CSR1
139      ;
140      000302      012746      140000      MOV.      #Q$MNC,-(SP)      ;CLEAR MRP NO-CLOCK BITS
141      000306      012746      041000      MOV.      #<Q$MSC+Q$LBD>,-(SP)      ;SET MRP CLOCK
142      000312      CALL      CSR1
143      ;
144      ;      DE-SELECTION
145      ;
146      000316      012746      001001      MOV.      #<Q$LBD+Q$LBP>,-(SP)      ;CLR DRIVE AND PULSE
147      000322      052716      000360      BIS.      #<Q$CSEL>,(SP)      ;CLR SELECTION BITS
148      000326      012746      176000      MOV.      #<Q$NCLK>,-(SP)      ;SET NO-CLOCKS
149      000332      CALL      CSR1
150      ;
151      000336      011666      000002      MOV.      (SP),2(SP)      ;MOVE RETURN ADDRESS DOWN STACK
152      000342      005726      TST.      (SP)+      ;POINT TO RETURN ADDRESS
153      000344      RETURN

```

```
155      ;
156      ;
157      ; DATA TRANSFER TO LOD BUS REG FROM MRP
158      ;
159      ; OUTPUT:
160      ; (SP) DATA FROM PRE-SELECTED MRP SOURCE
161      ;
162      ; MRPLB::
163      ;
164      ; MOV. #<Q$LBD+Q$LBP>,-(SP) ;CLR DRIVE AND PULSE
165      ; BIS. #<Q$CSEL>,(SP) ;CLR SELECTION BITS
166      ; MOV. #<Q$NCLK>,-(SP) ;SET NO-CLOCKS
167      ; BIS. #Q$MRP2,(SP) ;SOURCE IS MRP
168      ; CALL CSR1
169      ;
170      ; MOV. (SP),-(SP) ;MOVE RETURN ADDR UP STACK
171      ; MOV. QR$LBR,2(SP) ;MOVE DATA ONTO STACK
172      ;
173      ; MOV. #Q$MRP2,-(SP) ;CLEAR MRP SELECT
174      ; CLR. -(SP) ;SET NOTHING
175      ; CALL CSR1
176      ; RETURN
```

```
176      ;
177      ;
178      ;      MRP CONTROL REGISTER LOADING
179      ;
180      ;      INPUT:
181      ;      2(SP)      BIT SETTING FOR MRP CONTROL REGISTER
182      ;
183      000416      MRPCR::
184      000416      052766      100000      000002      BIS      #MNOBREG,2(SP)      ;INHIBIT BRANCH REGISTER SELECT
185      000424      MRPCRA::
186      000424      016667      000002      176424      MOV      2(SP),QR#LBR      ;CONTROL BITS DESTINED FOR MRP
187      000432      012746      001001      MOV      *(<Q#LBD+Q#LBP>,-(SP)      ;CLEAR DRIVE AND PULSE
188      000436      052716      000360      BIS      *(<Q#CSEL>,(SP)      ;CLR SELECTION BITS
189      000442      012746      000040      MOV      #Q#MRP,-(SP)      ;SELECT MRP
190      000446      CALL      CSR1
191      ;
192      000452      005046      CLR      -(SP)      ;CLEAR NOTHING
193      000454      012746      000001      MOV      #Q#LBP,-(SP)      ;SET PULSE
194      000460      CALL      CSR1
195      ;
196      000464      012746      000041      MOV      *(<Q#MRP+Q#LBP>,-(SP)      ;CLEAR CR SELECTION AND PULSE
197      000470      005045      CLR      -(SP)      ;SET NOTHING
198      000472      CALL      CSR1
199      ;
200      000476      011666      000002      MOV      (SP),2(SP)      ;MOVE RETURN ADDRESS DOWN STACK
201      000502      005726      TST      (SP)+      ;POINT TO RETURN ADDRESS
202      000504      RETURN
203
204      000001      .END
```


ALUCKE = 040006
ALUOE = 004006
A01 = 010000
BITVAL = 000000
BIT0 = 000001
BIT1 = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2 = 000004
BIT3 = 000010
BIT4 = 000020
BIT5 = 000040
BIT6 = 000100
BIT7 = 000200
BIT8 = 000400
BIT9 = 001000
BYTE0 = 000000
BYTE1 = 000001
BYTE10 = 000012
BYTE11 = 000013
BYTE12 = 000014
BYTE13 = 000015
BYTE14 = 000016
BYTE15 = 000017
BYTE16 = 000020
BYTE17 = 000021
BYTE18 = 000022
BYTE19 = 000023
BYTE2 = 000002
BYTE20 = 000024
BYTE21 = 000025
BYTE22 = 000026
BYTE23 = 000027
BYTE24 = 000030
BYTE25 = 000031
BYTE26 = 000032
BYTE27 = 000033
BYTE28 = 000034
BYTE29 = 000035
BYTE3 = 000003
BYTE30 = 000036
BYTE31 = 000037
BYTE32 = 000040
BYTE33 = 000041
BYTE34 = 000042
BYTE35 = 000043
BYTE36 = 000044
BYTE37 = 000045
BYTE38 = 000046
BYTE39 = 000047
BYTE4 = 000004
BYTE40 = 000050
BYTE41 = 000051
BYTE42 = 000052
BYTE43 = 000053
BYTE44 = 000054
BYTE45 = 000055
BYTE46 = 000056
BYTE47 = 000057
BYTE48 = 000060
BYTE49 = 000061
BYTE5 = 000005
BYTE50 = 000062
BYTE51 = 000063
BYTE52 = 000064
BYTE53 = 000065
BYTE54 = 000066
BYTE55 = 000067
BYTE56 = 000070
BYTE57 = 000071
BYTE58 = 000072
BYTE59 = 000073
BYTE6 = 000006
BYTE60 = 000074
BYTE61 = 000075
BYTE62 = 000076
BYTE63 = 000077
BYTE64 = 000100
BYTE65 = 000101
BYTE66 = 000102
BYTE67 = 000103
BYTE68 = 000104
BYTE69 = 000105
BYTE7 = 000007
BYTE70 = 000106
BYTE71 = 000107
BYTE72 = 000110
BYTE73 = 000111
BYTE74 = 000112
BYTE75 = 000113
BYTE76 = 000114
BYTE77 = 000115
BYTE78 = 000116
BYTE79 = 000117
BYTE8 = 000010
BYTE80 = 000120
BYTE81 = 000121
BYTE82 = 000122
BYTE83 = 000123
BYTE84 = 000124
BYTE85 = 000125
BYTE86 = 000126
BYTE87 = 000127
BYTE88 = 000130
BYTE89 = 000131
BYTE9 = 000011
BYTE90 = 000132
BYTE91 = 000133
BYTE92 = 000134
BYTE93 = 000135
BYTE94 = 000136
BYTE95 = 000137
BYTE96 = 000140
BYTE97 = 000141
BYTE98 = 000142
BYTE99 = 000143
BYTVAL = 000144
CBKALL = 001000
CBKCLK = 000400
CNOBRE = 100000
CPCCEN = 010000
CPREAD = 040000
CPWRTE = 020000
CSADRD = 000004
CSEQCI = 100000
CSOE = 000040
CSR1 = ***** GX
CSWRTE = 000100
DBR:RD = 000001
DB\$CPP = 001457
DB\$SPT = 000026
DB\$TPC = 000023
DISPCS = 100000
DMAAUR = 000005
DMARD = 000003
DMARUR = 000004
ENBR = 010000
LBMRP = 000152RG
LBMSC = 000250RG
LOC:EN = 000100
LOC:UA = 040000
LOC:UB = 100000
MAREN1 = 000001
MAREN2 = 004000
MARLOD = 010000
MAROUT = 000002
MAR:LO = 002000
MAR:OU = 000040
MBKALL = 001000
MBKCLK = 000400
MMADDR = 000100
MMLEFT = 000002
MMOE = 000004
MMWRTE = 000010
MNOBRE = 100000
MREN1 = 000001
MREN2 = 020000
MRPCR = 000416RG
MRPCRA = 000424RG
MRPLB = 000346RG
MRPSUB = 000000R
MSYN = 000040
N = 000144
PLB = 000010
PLC = 000020
PLD = 000030
PLWR = 000200
PLR:EN = 000200
QR\$CR1 = 176420
QR\$CR2 = 176422
QR\$LBR = 176424
QR\$ATTN = 000100
Q\$BCL = 000001
Q\$CCCP = 000040
Q\$CHB = 000400
Q\$CHRL = 000200
Q\$CLR = 000040
Q\$CNC = 030000
Q\$CP = 000060
Q\$CPCC = 000010
Q\$CP2 = 000260
Q\$CSC = 010000
Q\$CSEL = 000360
Q\$CSET = 000002
Q\$CSP = 020000
Q\$DMA = 000001
Q\$ENBK = 040000
Q\$ENOP = 020000
Q\$FAL = 004000
Q\$FC = 000045
Q\$FO = 000044
Q\$FP = 000046
Q\$HBF = 000002
Q\$ICP = 000006
002: Q\$IHB = 000003
002: Q\$IHRL = 000002
Q\$IMRP = 000007
Q\$LBD = 001000
Q\$LBDP = 001001
Q\$LBP = 000001
Q\$LDCD = 000003
Q\$LDMD = 000004
Q\$LDPP = 002000
Q\$LHP = 010000
Q\$MNC = 140000
Q\$MR = 000052
Q\$MRP = 000040
Q\$MRP2 = 000240
Q\$MSC = 040000
Q\$MSET = 000004
Q\$MSR = 100000
Q\$NCLK = 176000
Q\$PP = 000100
Q\$PPSW = 000320
002: Q\$PP2 = 000300
002: Q\$QHLT = 000013
002: Q\$QL = 000043
002: Q\$QLA = 000053
Q\$QLB = 000054
Q\$QLR = 000001
Q\$QW = 000042
Q\$RDCD = 000005
Q\$RDM = 000006
Q\$REFK = 001000
Q\$RNC = 006000
Q\$RSC = 004000
Q\$RSET = 000010
Q\$SM = 100000
Q\$SP = 000120
Q\$SP2 = 000340
RGD:EN = 000200
RGD:VA = 020000
SEQMM = 000000RG
SEQ:CI = 000010
S\$CLR = 000000
S\$LA = 000001
S\$QB = 000005
S\$QR = 000006
S\$QX = 000004
S\$SR = 000007
S\$S1 = 000010
S\$S2 = 000014
TD\$CTR = 176370
TD\$CTW = 176360
TD\$INL = 004000
TD\$MEM = 000270
TD\$OAR = 176344
TD\$OTR = 176346
TD\$QRD = 000274
TD\$SW = 176376
TD\$TAR = 176372
TD\$TAW = 176362
TD\$TDR = 176374
TD\$TDW = 176364
T\$AD = 000020
T\$BA = 000002
T\$BD = 000010
T\$BSO = 100000
T\$BT = 000020
T\$BTAR = 000030
T\$BTD = 002000
T\$CD = 000100
T\$CLK = 002000
T\$DISK = 000200
T\$DRD = 000004
T\$MEM = 010000
T\$FSA = 000000
T\$FSAB = 000004
T\$FSAC = 000014
T\$FSB2 = 000010
T\$IB = 000026
T\$IBAR = 000024
T\$IBE = 020000
T\$IBF = 040000
T\$ICD = 000040
T\$MODE = 004000
T\$OB = 000036
T\$OBE = 004000
T\$OBF = 010000
T\$OBRA = 000034
T\$OBWA = 000032

MRPSUB: MACRO: n110 27-MAR-80 15:12 PAGE: 10-2
SYMBOL TABLE

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

T\$OUTA= 100000	WORD18= 000044	WORD4 = 000010	WORD61= 000172	WORD82= 000244
T\$RBD0= 000200	WORD19= 000046	WORD40= 000120	WORD62= 000174	WORD83= 000246
T\$RNB= 000040	WORD2= 000004	WORD41= 000122	WORD63= 000176	WORD84= 000250
T\$RESET= 000000	WORD20= 000050	WORD42= 000124	WORD64= 000200	WORD85= 000252
T\$SC= 000022	WORD21= 000052	WORD43= 000126	WORD65= 000202	WORD86= 000254
T\$SCLK= 020000	WORD22= 000054	WORD44= 000130	WORD66= 000204	WORD87= 000256
T\$SEG1= 000000	WORD23= 000056	WORD45= 000132	WORD67= 000206	WORD88= 000260
T\$SEG2= 000001	WORD24= 000060	WORD46= 000134	WORD68= 000210	WORD89= 000262
T\$SEG3= 000002	WORD25= 000062	WORD47= 000136	WORD69= 000212	WORD9 = 000022
T\$SO= 000001	WORD26= 000064	WORD48= 000140	WORD7 = 000016	WORD90= 000264
T\$UBUS= 100000	WORD27= 000066	WORD49= 000142	WORD70= 000214	WORD91= 000266
T\$1CLK= 000400	WORD28= 000070	WORD5 = 000012	WORD71= 000216	WORD92= 000270
T\$BBEN= 000020	WORD29= 000072	WORD50= 000144	WORD72= 000220	WORD93= 000272
UBD.IN= 000020	WORD3 = 000006	WORD51= 000146	WORD73= 000222	WORD94= 000274
WORD0 = 000000	WORD30= 000074	WORD52= 000150	WORD74= 000224	WORD95= 000276
WORD1 = 000002	WORD31= 000076	WORD53= 000152	WORD75= 000226	WORD96= 000300
WORD10= 000024	WORD32= 000100	WORD54= 000154	WORD76= 000230	WORD97= 000302
WORD11= 000026	WORD33= 000102	WORD55= 000156	WORD77= 000232	WORD98= 000304
WORD12= 000030	WORD34= 000104	WORD56= 000160	WORD78= 000234	WORD99= 000306
WORD13= 000032	WORD35= 000106	WORD57= 000162	WORD79= 000236	WRDVAL= 000310
WORD14= 000034	WORD36= 000110	WORD58= 000164	WORD8 = 000020	WRTMM= 000066RG 002
WORD15= 000036	WORD37= 000112	WORD59= 000166	WORD80= 000240	XTREAD= 001000
WORD16= 000040	WORD38= 000114	WORD6 = 000014	WORD81= 000242	XTWRITE= 000400
WORD17= 000042	WORD39= 000116	WORD60= 000170		

. ABS. 000000 000
000000 001
MRPSUB: 000506 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3021 WORDS (12 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:43
MRPSUB, MRPSUB, SP=[20,1]IM,[20,1]MRPSUB

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

1      .TITLE--PPSUB--
2 000000 .PSECT--PPSUB--
3
4      .MCALL--UTSE$S,CLEF$S-
5
6      000003 EFN.3 = 3
7
8
9
10     .
11     .
12     .
13     .
14     .
15     .
16     .
17     .
18     .
19     .
20 000000 LBPP::
21 000000 016667 000002 176424 MOV. 2(SP),Q$RNC. ;MOVE DATA TO LOD-BUS-REG
22 000006 012746 001001 MOV. #<Q$LBD+Q$LBP>,-(SP) ;CLR-DRIVE-AND-PULSE
23 000012 052716 000360 BIS. #<Q$CSEL>,(SP) ;CLR-SELECTION-BITS
24 000016 012746 176000 MOV. #<Q$NCLK>,-(SP) ;SET-NO-CLOCKS
25 000022 052716 000300 BIS. #Q$PP2,(SP) ;SELECT-PP
26 000026 CALL CSR1 ;
27
28 000032 012746 006000 MOV. #Q$RNC,-(SP) ;CLEAR-CP-NO-CLOCK-BITS
29 000036 012746 001000 MOV. #Q$LBD,-(SP) ;SET-LOD-BUS-DRIVE
30 000042 CALL CSR1 ;
31
32
33     DE-SELECTION
34 000046 012746 001001 MOV. #<Q$LBD+Q$LBP>,-(SP) ;CLEAR-DRIVE-AND-PULSE
35 000052 052716 000360 BIS. #<Q$CSEL>,(SP) ;CLR-SELECTION-BITS
36 000056 012746 176000 MOV. #<Q$NCLK>,-(SP) ;SET-NO-CLOCKS
37 000062 CALL CSR1 ;
38
39 000066 011666 000002 MOV. (SP),2(SP) ;MOVE-RETURN-ADDRESS-DOWN-STACK
40 000072 005726 TST. (SP)+ ;POINT-TO-RETURN-ADDRESS
41 000074 RETURN

```

```

43      ;
44      ;
45      ; DATA TRANSFER
46      ; LOD BUS REGISTER TO A DESTINATION ON THE PPS BUS
47      ; SINGLE CLOCK SEQUENCER ONLY
48      ;
49      ; INPUT:
50      ; 2(SP) DATA FOR PRE-SELECTED PPS DESTINATION
51      ;
52      ;
53      LBPSC:
54      000076 016667 000002 176424 MOV 2(SP),Q$LBR ;MOVE DATA TO LOD BUS REG
55      000104 012746 001001 MOV #<Q$LBD+Q$LBP>,-(SP) ;CLEAR DRIVE AND PULSE
56      000110 052716 000360 BIS #Q$CSEL,(SP) ;CLR SELECTION BITS
57      000114 012746 176000 MOV #<Q$NCLK>,-(SP) ;SET NO CLOCKS
58      000120 052716 000300 BIS #Q$PP2,(SP) ;SELECT PP
59      000124 CALL CSR1 ;WRITE CONTROL REGISTER
60      ;
61      000130 012746 006000 MOV #Q$RNC,-(SP) ;CLEAR PP NO-CLOCK BITS
62      000134 012746 005000 MOV #<Q$RSC+Q$LBD>,-(SP) ;SET PP CLOCK
63      000140 CALL CSR1 ;
64      ;
65      ; DE-SELECTION
66      ;
67      000144 012746 001001 MOV #<Q$LBD+Q$LBP>,-(SP) ;CLEAR DRIVE AND PULSE
68      000150 052716 000360 BIS #Q$CSEL,(SP) ;CLR SELECTION BITS
69      000154 012746 176000 MOV #<Q$NCLK>,-(SP) ;SET NO CLOCKS
70      000160 CALL CSR1 ;
71      ;
72      000164 011666 000002 MOV (SP),2(SP) ;MOVE RETURN ADDRESS DOWN STACK
73      000170 005726 TST (SP)+ ;POINT TO RETURN ADDRESS
74      000172 RETURN

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

76      :
77      :
78      :      DATA TRANSFER TO LOD BUS REG FROM PPS.
79      :
80      :      OUTPUT:
81      :      (SP) DATA FROM PRE-SELECTED PPS SOURCE.
82      :
83      :
84 000174      PPLB::
85 000174      012746 001001      MOV.    #<Q$LBD+Q$LBP>,-(SP)      ;CLEAR DRIVE AND PULSE
86 000200      052716 000350      BIS.    #Q$CSEL,(SP)      ;CLR SELECTION BITS
87 000204      012746 176000      MOV.    #<Q$NCLK>,-(SP)      ;SET NO-CLOCKS
88 000210      052716 000300      BIS.    #Q$PP2,(SP)      ;SOURCE IS PP
89 000214      CALL.    CSR1
90 000220      011646      MOV.    (SP),-(SP)      ;MOVE RETURN ADDR UP STACK
91 000222      016766 176424 000002      MOV.    QR$LBR,2(SP)      ;MOVE DATA ONTO STACK
92      :
93 000230      012746 000300      MOV.    #Q$PP2,-(SP)      ;CLEAR PP SELECT
94 000234      005046      CLR.    -(SP)      ;SET NOTHING
95 000236      CALL.    CSR1
96 000242      RETURN.

```

```
98
99
100
101
102
103
104
105
106 000244
107 000244 016667 000002 176424
108 000252 012746 001001
109 000256 052716 000360
110 000262 012746 000100
111 000266
112
113 000272 005046
114 000274 012746 000001
115 000300
116
117 000304 012746 000101
118 000310 005046
119 000312
120
121 000316 011666 000002
122 000322 005726
123 000324

;
;
; PPS CONTROL REGISTER LOADING
;
; INPUT:
; 2(SP) BIT SETTING FOR PPS CONTROL REGISTER
;
;
PPCR::
MOV 2(SP),QR$LBR ;CONTROL BITS DESTINED FOR PPS
MOV #(<Q$LBD+Q$LBP>,-(SP) ;CLEAR DRIVE AND PULSE
BIS #Q$CSEL,(SP) ;CLR SELECTION BITS
MOV #Q$PP,-(SP) ;SELECT PP
CALL CSR1
;
;
CLR -(SP) ;CLEAR NOTHING
MOV #Q$LBP,-(SP) ;SET PULSE
CALL CSR1
;
;
MOV #(<Q$PP+Q$LBP>,-(SP) ;CLEAR CR SELECTION AND PULSE
CLR -(SP) ;SET NOTHING
CALL CSR1
;
;
MOV (SP),2(SP) ;MOVE RETURN ADDRESS DOWN STACK
TST (SP)+ ;POINT TO RETURN ADDRESS
RETURN
```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

125      ;
126      ;
127      ;
128      ;
129      ;
130      ;
131      ;
132      ;
133      ;
134      ;
135      ;
136      000326      SELPG::
137      000326      005046      CLR.      -(SP)      ;START MICROCODE AT 0
138      000330      CALL.     SEQCS      ;
139      000334      005046      CLR.      -(SP)      ;RESET BR INHIBIT
140      000336      CALL.     CPCR      ;
141      000342      012746      000377      MOV.     #377,-(SP)      ;SET MRP MICRO ADDRESS = X'FF' (JUMP SELF)
142      000346      CALL.     SEQMM      ;
143      000352      005046      CLR.      -(SP)      ;RESET BR INHIBIT
144      000354      CALL.     MRPCR      ;
145      000360      012767      001000      176422      MOV.     #Q$REBK,QR$CR2      ;RE-ARM INTERRUPTS
146      000366      012767      120000      176422      MOV.     *(Q$SM+Q$ENOP),QR$CR2      ;SET SEARCH MODE + ENABLE INTERRUPTS
147      000374      012746      000360      MOV.     #Q$CSEL,-(SP)      ;CLEAR ALL SELECTIONS
148      000400      052716      001001      BIS.     *(Q$LBD+Q$LBP),(SP)      ;CLEAR DRIVE AND PULSE
149      000404      052716      030000      BIS.     #Q$CNC,(SP)      ;CLEAR CP NO-CLOCK
150      000410      005046      CLR.      -(SP)      ;SET NOTHING
151      000412      CALL.     CSR1      ;
152      ;
153      000416      012767      000003      176424      MOV.     #Q$LCD,QR$LBR      ;MOVE ATTN CODE TO LOD BUS REG
154      000424      012767      120100      176422      MOV.     *(Q$ATTN+Q$SM+Q$ENOP),QR$CR2      ;SET ATTN CODE READY
155      000432      016701      176422      1$:      MOV.     QR$CR2,R1      ;READ CSR2
156      000436      032701      000100      BIT.     #Q$ATTN,R1      ;ATTN CLEAR
157      000442      001373      BNE.     1$      ;NO READ AGAIN
158      ;
159      000444      012767      000041      176424      MOV.     #041,QR$LBR      ;CD MEMORY START ADDRESS = X'21'
160      000452      012767      120040      176422      MOV.     *(Q$CCCP+Q$SM+Q$ENOP),QR$CR2      ;SET CC TO CP
161      000460      016701      176422      2$:      MOV.     QR$CR2,R1      ;READ CSR2
162      000464      032701      000040      BIT.     #Q$CCCP,R1      ;IS CC TO CP CLEAR
163      000470      001373      BNE.     2$      ;NO READ AGAIN
164      ;
165      000472      012767      000001      176424      MOV.     #1,QR$LBR      ;TRANSFER COUNT = 1 WORD
166      000500      012767      120040      176422      MOV.     *(Q$CCCP+Q$SM+Q$ENOP),QR$CR2      ;SET CC TO CP
167      000506      016701      176422      3$:      MOV.     QR$CR2,R1      ;READ CSR2
168      000512      032701      000040      BIT.     #Q$CCCP,R1      ;IS CC TO CP CLEAR
169      000516      001373      BNE.     3$      ;NO READ AGAIN
170      ;
171      000520      016667      000002      000000G      MOV.     2(SP),DATA1      ;PUT PAGE VALUE INTO BUFFER
172      000526      005367      000000G      DEC.     DATA1      ;MICROCODE REINCREMENTS
173      000532      012767      000000G      176424      MOV.     #DATA1,QR$LBR      ;CC MEMORY DATA BUFFER
174      000540      012767      120040      176422      MOV.     *(Q$CCCP+Q$SM+Q$ENOP),QR$CR2      ;SET CC TO CP
175      ;
176      ;
177      ;
178      000546      WAIT FOR INTERRUPT FROM CP
179      ;
180      000560      WTSE$. #EFN.3
181      ;
182      ;
183      ;
184      ;
185      ;
186      ;
187      ;
188      ;
189      ;
190      ;
191      ;
192      ;
193      ;
194      ;
195      ;
196      ;
197      ;
198      ;
199      ;
200      ;
201      ;
202      ;
203      ;
204      ;
205      ;
206      ;
207      ;
208      ;
209      ;
210      ;
211      ;
212      ;
213      ;
214      ;
215      ;
216      ;
217      ;
218      ;
219      ;
220      ;
221      ;
222      ;
223      ;
224      ;
225      ;
226      ;
227      ;
228      ;
229      ;
230      ;
231      ;
232      ;
233      ;
234      ;
235      ;
236      ;
237      ;
238      ;
239      ;
240      ;
241      ;
242      ;
243      ;
244      ;
245      ;
246      ;
247      ;
248      ;
249      ;
250      ;
251      ;
252      ;
253      ;
254      ;
255      ;
256      ;
257      ;
258      ;
259      ;
260      ;
261      ;
262      ;
263      ;
264      ;
265      ;
266      ;
267      ;
268      ;
269      ;
270      ;
271      ;
272      ;
273      ;
274      ;
275      ;
276      ;
277      ;
278      ;
279      ;
280      ;
281      ;
282      ;
283      ;
284      ;
285      ;
286      ;
287      ;
288      ;
289      ;
290      ;
291      ;
292      ;
293      ;
294      ;
295      ;
296      ;
297      ;
298      ;
299      ;
300      ;
301      ;
302      ;
303      ;
304      ;
305      ;
306      ;
307      ;
308      ;
309      ;
310      ;
311      ;
312      ;
313      ;
314      ;
315      ;
316      ;
317      ;
318      ;
319      ;
320      ;
321      ;
322      ;
323      ;
324      ;
325      ;
326      ;
327      ;
328      ;
329      ;
330      ;
331      ;
332      ;
333      ;
334      ;
335      ;
336      ;
337      ;
338      ;
339      ;
340      ;
341      ;
342      ;
343      ;
344      ;
345      ;
346      ;
347      ;
348      ;
349      ;
350      ;
351      ;
352      ;
353      ;
354      ;
355      ;
356      ;
357      ;
358      ;
359      ;
360      ;
361      ;
362      ;
363      ;
364      ;
365      ;
366      ;
367      ;
368      ;
369      ;
370      ;
371      ;
372      ;
373      ;
374      ;
375      ;
376      ;
377      ;
378      ;
379      ;
380      ;
381      ;
382      ;
383      ;
384      ;
385      ;
386      ;
387      ;
388      ;
389      ;
390      ;
391      ;
392      ;
393      ;
394      ;
395      ;
396      ;
397      ;
398      ;
399      ;
400      ;
401      ;
402      ;
403      ;
404      ;
405      ;
406      ;
407      ;
408      ;
409      ;
410      ;
411      ;
412      ;
413      ;
414      ;
415      ;
416      ;
417      ;
418      ;
419      ;
420      ;
421      ;
422      ;
423      ;
424      ;
425      ;
426      ;
427      ;
428      ;
429      ;
430      ;
431      ;
432      ;
433      ;
434      ;
435      ;
436      ;
437      ;
438      ;
439      ;
440      ;
441      ;
442      ;
443      ;
444      ;
445      ;
446      ;
447      ;
448      ;
449      ;
450      ;
451      ;
452      ;
453      ;
454      ;
455      ;
456      ;
457      ;
458      ;
459      ;
460      ;
461      ;
462      ;
463      ;
464      ;
465      ;
466      ;
467      ;
468      ;
469      ;
470      ;
471      ;
472      ;
473      ;
474      ;
475      ;
476      ;
477      ;
478      ;
479      ;
480      ;
481      ;
482      ;
483      ;
484      ;
485      ;
486      ;
487      ;
488      ;
489      ;
490      ;
491      ;
492      ;
493      ;
494      ;
495      ;
496      ;
497      ;
498      ;
499      ;
500      ;
501      ;
502      ;
503      ;
504      ;
505      ;
506      ;
507      ;
508      ;
509      ;
510      ;
511      ;
512      ;
513      ;
514      ;
515      ;
516      ;
517      ;
518      ;
519      ;
520      ;
521      ;
522      ;
523      ;
524      ;
525      ;
526      ;
527      ;
528      ;
529      ;
530      ;
531      ;
532      ;
533      ;
534      ;
535      ;
536      ;
537      ;
538      ;
539      ;
540      ;
541      ;
542      ;
543      ;
544      ;
545      ;
546      ;
547      ;
548      ;
549      ;
550      ;
551      ;
552      ;
553      ;
554      ;
555      ;
556      ;
557      ;
558      ;
559      ;
560      ;
561      ;
562      ;
563      ;
564      ;
565      ;
566      ;
567      ;
568      ;
569      ;
570      ;
571      ;
572      ;
573      ;
574      ;
575      ;
576      ;
577      ;
578      ;
579      ;
580      ;
581      ;
582      ;
583      ;
584      ;
585      ;
586      ;
587      ;
588      ;
589      ;
590      ;
591      ;
592      ;
593      ;
594      ;
595      ;
596      ;
597      ;
598      ;
599      ;
600      ;
601      ;
602      ;
603      ;
604      ;
605      ;
606      ;
607      ;
608      ;
609      ;
610      ;
611      ;
612      ;
613      ;
614      ;
615      ;
616      ;
617      ;
618      ;
619      ;
620      ;
621      ;
622      ;
623      ;
624      ;
625      ;
626      ;
627      ;
628      ;
629      ;
630      ;
631      ;
632      ;
633      ;
634      ;
635      ;
636      ;
637      ;
638      ;
639      ;
640      ;
641      ;
642      ;
643      ;
644      ;
645      ;
646      ;
647      ;
648      ;
649      ;
650      ;
651      ;
652      ;
653      ;
654      ;
655      ;
656      ;
657      ;
658      ;
659      ;
660      ;
661      ;
662      ;
663      ;
664      ;
665      ;
666      ;
667      ;
668      ;
669      ;
670      ;
671      ;
672      ;
673      ;
674      ;
675      ;
676      ;
677      ;
678      ;
679      ;
680      ;
681      ;
682      ;
683      ;
684      ;
685      ;
686      ;
687      ;
688      ;
689      ;
690      ;
691      ;
692      ;
693      ;
694      ;
695      ;
696      ;
697      ;
698      ;
699      ;
700      ;
701      ;
702      ;
703      ;
704      ;
705      ;
706      ;
707      ;
708      ;
709      ;
710      ;
711      ;
712      ;
713      ;
714      ;
715      ;
716      ;
717      ;
718      ;
719      ;
720      ;
721      ;
722      ;
723      ;
724      ;
725      ;
726      ;
727      ;
728      ;
729      ;
730      ;
731      ;
732      ;
733      ;
734      ;
735      ;
736      ;
737      ;
738      ;
739      ;
740      ;
741      ;
742      ;
743      ;
744      ;
745      ;
746      ;
747      ;
748      ;
749      ;
750      ;
751      ;
752      ;
753      ;
754      ;
755      ;
756      ;
757      ;
758      ;
759      ;
760      ;
761      ;
762      ;
763      ;
764      ;
765      ;
766      ;
767      ;
768      ;
769      ;
770      ;
771      ;
772      ;
773      ;
774      ;
775      ;
776      ;
777      ;
778      ;
779      ;
780      ;
781      ;
782      ;
783      ;
784      ;
785      ;
786      ;
787      ;
788      ;
789      ;
790      ;
791      ;
792      ;
793      ;
794      ;
795      ;
796      ;
797      ;
798      ;
799      ;
800      ;
801      ;
802      ;
803      ;
804      ;
805      ;
806      ;
807      ;
808      ;
809      ;
810      ;
811      ;
812      ;
813      ;
814      ;
815      ;
816      ;
817      ;
818      ;
819      ;
820      ;
821      ;
822      ;
823      ;
824      ;
825      ;
826      ;
827      ;
828      ;
829      ;
830      ;
831      ;
832      ;
833      ;
834      ;
835      ;
836      ;
837      ;
838      ;
839      ;
840      ;
841      ;
842      ;
843      ;
844      ;
845      ;
846      ;
847      ;
848      ;
849      ;
850      ;
851      ;
852      ;
853      ;
854      ;
855      ;
856      ;
857      ;
858      ;
859      ;
860      ;
861      ;
862      ;
863      ;
864      ;
865      ;
866      ;
867      ;
868      ;
869      ;
870      ;
871      ;
872      ;
873      ;
874      ;
875      ;
876      ;
877      ;
878      ;
879      ;
880      ;
881      ;
882      ;
883      ;
884      ;
885      ;
886      ;
887      ;
888      ;
889      ;
890      ;
891      ;
892      ;
893      ;
894      ;
895      ;
896      ;
897      ;
898      ;
899      ;
900      ;
901      ;
902      ;
903      ;
904      ;
905      ;
906      ;
907      ;
908      ;
909      ;
910      ;
911      ;
912      ;
913      ;
914      ;
915      ;
916      ;
917      ;
918      ;
919      ;
920      ;
921      ;
922      ;
923      ;
924      ;
925      ;
926      ;
927      ;
928      ;
929      ;
930      ;
931      ;
932      ;
933      ;
934      ;
935      ;
936      ;
937      ;
938      ;
939      ;
940      ;
941      ;
942      ;
943      ;
944      ;
945      ;
946      ;
947      ;
948      ;
949      ;
950      ;
951      ;
952      ;
953      ;
954      ;
955      ;
956      ;
957      ;
958      ;
959      ;
960      ;
961      ;
962      ;
963      ;
964      ;
965      ;
966      ;
967      ;
968      ;
969      ;
970      ;
971      ;
972      ;
973      ;
974      ;
975      ;
976      ;
977      ;
978      ;
979      ;
980      ;
981      ;
982      ;
983      ;
984      ;
985      ;
986      ;
987      ;
988      ;
989      ;
990      ;
991      ;
992      ;
993      ;
994      ;
995      ;
996      ;
997      ;
998      ;
999      ;
1000      ;

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

182.      ;      RE-ARM INTERRUPTS.
183.      ;
184 000572 012767 100400 176422 MOV.  *(<Q$SM+Q$CHB>,QR$CR2.  ;CLEAR INTERRUPT (USE HIT BUFFER INT)
185 000600 012767 101000 176422 MOV.  *(<Q$SM+Q$REBK>,QR$CR2.  ;RE-ARM
186 000606 012767 160000 176422 MOV.  *(<Q$SM+Q$ENBK+Q$ENOP>,QR$CR2. ;ENABLE
187      ;
188 000614 005046      CLR.  -(SP)      ;CLEAR NOTHING IN CSR1
189 000616 012746 176000 MOV.  *Q$NCLK,-(SP) ;SET NO-CLOCKS
190 000622      CALL.  CSR1
191 000626 005067 176422 CLR.  QR$CR2.  ;SET LOAD MODE
192.      ;
193 000632 012746 001761 MOV.  *1761,-(SP)      ;START CP AT X'3F1'
194 000636      CALL.  SEQCS.
195 000642 005046      CLR.  -(SP)      ;RESET BR INHIBIT
196 000644      CALL.  CPCR
197 000650 012746 000377 MOV.  *377,-(SP)      ;SET MRP MICRO ADDRESS = X'FF' (JUMP SELF)
198 000654      CALL.  SEQMM.
199 000660 005046      CLR.  -(SP)      ;RESET BR INHIBIT
200 000662      CALL.  MRPCR.
201 000666 012767 001000 176422 MOV.  *Q$REBK,QR$CR2  ;RE-ARM INTERRUPTS
202 000674 012767 120000 176422 MOV.  *(<Q$SM+Q$ENOP>,QR$CR2.  ;SET SEARCH MODE + ENABLE INTERRUPTS
203 000702 012746 000360 MOV.  *Q$CSEL,-(SP)  ;CLEAR ALL SELECTIONS
204 000706 052716 001001 BIS.  *(<Q$LBD+Q$LBP>,(SP) ;CLEAR DRIVE AND PULSE
205 000712 052716 036000 BIS.  *(<Q$RNC+Q$CNC>,(SP) ;CLEAR CP AND PPS NO-CLOCKS
206 000716 005046      CLR.  -(SP)      ;SET NOTHING
207 000720      CALL.  CSR1
208      ;
209      ;      WAIT FOR A TIME.
210      ;
211 000724 012701 000144 MOV.  #100,R1      ;LOOP 100 TIMES
212 000730 005002      CLR.  R2.      ;ADD NOTHING
213 000732 000202 4$: ADD.  R2,R2.
214 000734 005301      DEC.  R1
215 000736 001375      BNE.  4$
216      ;
217 000740 005046      CLR.  -(SP)      ;CLEAR NOTHING IN CSR1
218 000742 012746 176000 MOV.  *Q$NCLK,-(SP) ;SET NO-CLOCKS
219 000746      CALL.  CSR1
220 000752 005067 176422 CLR.  QR$CR2.  ;SET LOAD MODE
221      ;
222 000756 011666 000002 MOV.  (SP),2(SP)      ;MOVE RETURN ADDRESS DOWN STACK
223 000762 005726      TST.  (SP)+      ;POINT TO RETURN ADDRESS
224 000764      RETURN.
225      ;
226      000001      .END.

```


ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	PLR.EN = 000200	Q\$RDMD = 000006
ALUOE = 004000	BYTE43 = 000053	BYTE95 = 000137	PPCR = 000244RG	002.Q\$REBK = 001000
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	PPLB = 000174RG	002.Q\$RNC = 006000
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	Q\$CR1 = 176420	Q\$RSC = 004000
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	Q\$CR2 = 176422	Q\$RSET = 000010
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	Q\$LBR = 176424	Q\$SM = 100000
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	Q\$HTTN = 000100	Q\$SP = 000120
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	Q\$BCL = 000001	Q\$SP2 = 000340
BIT12 = 010000	BYTE5 = 000005	CBKCLK = 000400	Q\$CCCP = 000040	RGQ.EN = 000200
BIT13 = 020000	BYTE50 = 000062	CNOBRE = 100000	Q\$CHB = 000400	RGQ.VA = 020000
BIT14 = 040000	BYTE51 = 000063	CPCCEN = 010000	Q\$CHRL = 000200	SELPG = 000326RG
BIT15 = 100000	BYTE52 = 000064	CPCR = ***** GX	Q\$CLR = 000040	002.SEQCS = ***** GX
BIT2 = 000004	BYTE53 = 000065	CPREAD = 040000	Q\$CNC = 030000	SEQMM = ***** GX
BIT3 = 000010	BYTE54 = 000066	CPWRTE = 020000	Q\$CP = 000060	SEQ.CI = 000010
BIT4 = 000020	BYTE55 = 000067	CSABRD = 000004	Q\$CPCC = 000010	S\$CLR = 000000
BIT5 = 000040	BYTE56 = 000070	CSEQCI = 100000	Q\$CP2 = 000260	S\$LA = 000001
BIT6 = 000100	BYTE57 = 000071	CSOE = 000040	Q\$CSC = 010000	S\$QB = 000005
BIT7 = 000200	BYTE58 = 000072	CSR1 = ***** GX	Q\$CSEL = 000360	S\$QR = 000006
BIT8 = 000400	BYTE59 = 000073	CSWRTE = 000100	Q\$CSET = 000002	S\$OX = 000004
BIT9 = 001000	BYTE6 = 000006	DATA1 = ***** GX	Q\$CSP = 020000	S\$SR = 000007
BYTE0 = 000000	BYTE60 = 000074	DBR.RD = 000001	Q\$DMA = 000001	S\$S1 = 000010
BYTE1 = 000001	BYTE61 = 000075	DB\$CPP = 001457	Q\$ENBK = 040000	S\$S2 = 000014
BYTE10 = 000012	BYTE62 = 000076	DB\$SPT = 000026	Q\$ENOP = 020000	TD\$CTR = 176370
BYTE11 = 000013	BYTE63 = 000077	DB\$TPC = 000023	Q\$FAL = 004000	TD\$CTU = 176360
BYTE12 = 000014	BYTE64 = 000100	DISPGS = 100000	Q\$FC = 000045	TD\$INL = 004000
BYTE13 = 000015	BYTE65 = 000101	DMAWR = 000005	Q\$FO = 000044	TD\$MEM = 000270
BYTE14 = 000016	BYTE66 = 000102	DMARRD = 000003	Q\$FP = 000046	TD\$OR = 176344
BYTE15 = 000017	BYTE67 = 000103	DMARRD = 000004	Q\$HBF = 000002	TD\$OTR = 176346
BYTE16 = 000020	BYTE68 = 000104	EFN.3 = 000003	Q\$ICP = 000006	TD\$ORD = 000274
BYTE17 = 000021	BYTE69 = 000105	ENBR = 010000	Q\$IH = 000003	TD\$SW = 176376
BYTE18 = 000022	BYTE7 = 000007	LBPP = 000000RG	002.Q\$IHL = 000002	TD\$STAR = 176372
BYTE19 = 000023	BYTE70 = 000106	LBPPSC = 000076RG	002.Q\$IMRP = 000007	TD\$TAR = 176362
BYTE2 = 000002	BYTE71 = 000107	LOC.EN = 000100	Q\$LBD = 001000	TD\$TDR = 176374
BYTE20 = 000024	BYTE72 = 000110	LOC.WA = 040000	Q\$LBDP = 001001	TD\$TDW = 176364
BYTE21 = 000025	BYTE73 = 000111	LOC.WB = 100000	Q\$LBP = 000001	T\$AD = 000020
BYTE22 = 000026	BYTE74 = 000112	MAREN1 = 000001	Q\$LDCD = 000003	T\$BA = 000002
BYTE23 = 000027	BYTE75 = 000113	MAREN2 = 004000	Q\$LDMD = 000004	T\$BD = 000010
BYTE24 = 000030	BYTE76 = 000114	MARLOD = 010000	Q\$LDPP = 002000	T\$BSQ = 100000
BYTE25 = 000031	BYTE77 = 000115	MAROUT = 000002	Q\$LHP = 010000	T\$BT = 000020
BYTE26 = 000032	BYTE78 = 000116	MAR.LO = 002000	Q\$INC = 140000	T\$BTAR = 000030
BYTE27 = 000033	BYTE79 = 000117	MAR.OU = 000040	Q\$MR = 000052	T\$BTD = 002000
BYTE28 = 000034	BYTE8 = 000010	MBKALL = 001000	Q\$MRP = 000040	T\$CD = 000100
BYTE29 = 000035	BYTE80 = 000120	MBKCLK = 000400	Q\$MRP2 = 000240	T\$CLK = 002000
BYTE3 = 000003	BYTE81 = 000121	MMADRD = 000100	Q\$MSC = 040000	T\$DISK = 000200
BYTE30 = 000036	BYTE82 = 000122	MMLEFT = 000002	Q\$MSET = 000004	T\$DRD = 000004
BYTE31 = 000037	BYTE83 = 000123	MMOE = 000004	Q\$MSP = 100000	T\$MEM = 010000
BYTE32 = 000040	BYTE84 = 000124	MMWRTE = 000010	Q\$NCLK = 176000	T\$FSAR = 000000
BYTE33 = 000041	BYTE85 = 000125	MNOBRE = 100000	Q\$PP = 000100	T\$FSAB = 000004
BYTE34 = 000042	BYTE86 = 000126	MREN1 = 000001	Q\$PPSW = 000320	T\$FSAC = 000014
BYTE35 = 000043	BYTE87 = 000127	MREN2 = 020000	Q\$PP2 = 000300	T\$FSB2 = 000010
BYTE36 = 000044	BYTE88 = 000130	MRPCR = ***** GX	Q\$QHLT = 000013	T\$IB = 000026
BYTE37 = 000045	BYTE89 = 000131	MSYN = 000040	Q\$QL = 000043	T\$IBAR = 000024
BYTE38 = 000046	BYTE9 = 000011	N = 000144	Q\$QLA = 000053	T\$IBE = 020000
BYTE39 = 000047	BYTE90 = 000132	PLB = 000010	Q\$QLB = 000054	T\$IBF = 040000
BYTE4 = 000004	BYTE91 = 000133	PLC = 000020	Q\$QLR = 000001	T\$ICD = 000040
BYTE40 = 000050	BYTE92 = 000134	PLD = 000030	Q\$QW = 000042	T\$MODE = 004000
BYTE41 = 000051	BYTE93 = 000135	PLRUR = 000200	Q\$RCD = 000005	T\$OB = 000036

T#0BE = 000000	WORD15 = 000036	WORD37 = 000112	WORD59 = 000166	WORD80 = 000240
T#0BF = 010000	WORD16 = 000040	WORD38 = 000114	WORD6 = 000014	WORD81 = 000242
T#0BR = 000034	WORD17 = 000042	WORD39 = 000116	WORD60 = 000170	WORD82 = 000244
T#0BW = 000032	WORD18 = 000044	WORD4 = 000010	WORD61 = 000172	WORD83 = 000246
T#0UT = 100000	WORD19 = 000046	WORD40 = 000120	WORD62 = 000174	WORD84 = 000250
T#RBD = 000200	WORD2 = 000004	WORD41 = 000122	WORD63 = 000176	WORD85 = 000252
T#RNB = 000040	WORD20 = 000050	WORD42 = 000124	WORD64 = 000200	WORD86 = 000254
T#RSE = 040000	WORD21 = 000052	WORD43 = 000126	WORD65 = 000202	WORD87 = 000256
T#SC = 000022	WORD22 = 000054	WORD44 = 000130	WORD66 = 000204	WORD88 = 000260
T#SCLK = 020000	WORD23 = 000056	WORD45 = 000132	WORD67 = 000206	WORD89 = 000262
T#SEG1 = 000000	WORD24 = 000060	WORD46 = 000134	WORD68 = 000210	WORD9 = 000022
T#SEG2 = 000001	WORD25 = 000062	WORD47 = 000136	WORD69 = 000212	WORD90 = 000264
T#SEG3 = 000002	WORD26 = 000064	WORD48 = 000140	WORD7 = 000016	WORD91 = 000266
T#SO = 000001	WORD27 = 000066	WORD49 = 000142	WORD70 = 000214	WORD92 = 000270
T#UBUS = 100000	WORD28 = 000070	WORD5 = 000012	WORD71 = 000216	WORD93 = 000272
T#1CLK = 000400	WORD29 = 000072	WORD50 = 000144	WORD72 = 000220	WORD94 = 000274
T#BBEN = 000020	WORD3 = 000006	WORD51 = 000146	WORD73 = 000222	WORD95 = 000276
UBD, IN = 000020	WORD30 = 000074	WORD52 = 000150	WORD74 = 000224	WORD96 = 000300
WORD0 = 000000	WORD31 = 000076	WORD53 = 000152	WORD75 = 000226	WORD97 = 000302
WORD1 = 000002	WORD32 = 000100	WORD54 = 000154	WORD76 = 000230	WORD98 = 000304
WORD10 = 000024	WORD33 = 000102	WORD55 = 000156	WORD77 = 000232	WORD99 = 000306
WORD11 = 000026	WORD34 = 000104	WORD56 = 000160	WORD78 = 000234	WORDVAL = 000310
WORD12 = 000030	WORD35 = 000106	WORD57 = 000162	WORD79 = 000236	XTREAD = 001000
WORD13 = 000032	WORD36 = 000110	WORD58 = 000164	WORD8 = 000020	XTWRITE = 000400
WORD14 = 000034				

. ABS. 000000 000
 000000 001
 PPSUB: 000766 002
 ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3288 WORDS (13 PAGES)
 DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
 ELAPSED TIME: 00:00:46
 PPSUB, PPSUB/SP-C20, 1JIM, C20, 1JPPSUB

```
1
2
3 000000 .TITLE MRP.
4 .PSECT MRP.
5 .LIST MEB.
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
```

HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS
MATCH REPORT PROCESSOR ROUTINES.

PARSE THE COMMAND LINE AND TRANSFER CONTROL TO ONE OF THE
MRP SUB-MODULES OR BACK TO QMAIN. CONTROL COULD HAVE BEEN
PASSED TO MRP IN ONE OF THREE WAYS:

1. FROM QMAIN IF MRP WAS SELECTED AS THE FIRST PROCESSOR
UPON ENTRY TO THE PROGRAM.
MR>LD MD 0
2. FROM QMAIN IF MRP WAS SELECTED FROM ONE OF QMAIN'S
OTHER SUB-MODULES.
CP>MR LD MD 0
3. ON RETURN FROM ONE OF MRP'S SUB-MODULES
MR>LD MD 0

MRP PARSES AT THE SECOND LEVEL OF CONTROL (SEE NOTES AT
QMAIN ON LEVELS OF CONTROL). IN THE EXAMPLES ABOVE, MRP
WOULD PARSE 'LD' AND TRANSFER CONTROL TO THE MRP SUB-
MODULE MRLD. IF MRP SHOULD ENCOUNTER A STRING WHICH IS
NOT A VALID COMMAND MNEMONIC, MRP CONSIDERS THE STRING TO
BE A PROCESSOR MNEMONIC AND RETURNS CONTROL TO QMAIN. EG:
MR>CP LD CS 0
THE STRING 'CP' IS NOT A VALID MRP (SECOND LEVEL) COMMAND.
MRP RETURNS CONTROL TO QMAIN WHICH IN TURN WILL TRANSFER
CONTROL TO ITS SUB-MODULE CP.

MRP SUB-MODULES:

- MRLD LOAD MEMORIES.
- MRPR PRINT MEMORY CONTENTS.
- MRREST REMAINING COMMANDS (EXCEPT DEBUGGING)
- MRBUG DEBUGGING COMMANDS.

```
39      ;
40      ;
41      ;
42      ;
43      ;
44      ;
45      000000      ;
46      000000      101      124      ;
47      000002      000000G      ;
48      000004      117      106      ;
49      000006      000000G      ;
50      000010      123      123      ;
51      000012      000000G      ;
52      000014      107      117      ;
53      000016      000000G      ;
54      000020      123      124      ;
55      000022      000000G      ;
56      000024      122      105      ;
57      000026      000000G      ;
58      000030      114      104      ;
59      000032      000000G      ;
60      000034      120      122      ;
61      000036      000000G      ;
62      000040      122      123      ;
63      000042      000000G      ;
64      000044      103      114      ;
65      000046      000000G      ;
66      000012      ;
```

TABLE OF MRP COMMAND MNEMONICS AND THEIR ASSOCIATED ROUTINE ADDRESSES.

MRTBL:

.ASCII	/AT/	:SET BREAKPOINT.
.WORD	AT1	
.ASCII	/OF/	:DISABLE BREAKPOINT.
.WORD	OF1	
.ASCII	/SS/	:SINGLE STEP.
.WORD	SS1	
.ASCII	/GO/	:PROCEED FROM BREAKPOINT.
.WORD	GO1	
.ASCII	/ST/	:STORE INTO A REG.
.WORD	ST1	
.ASCII	/RE/	:READ FROM A REG.
.WORD	RE1	
.ASCII	/LD/	:LOAD MEMORY.
.WORD	LD1	
.ASCII	/PR/	:PRINT FROM MEMORY.
.WORD	PR1	
.ASCII	/RS/	:RESET.
.WORD	RS1	
.ASCII	/CL/	:CALL OR LOADER.
.WORD	CL1	

MRNUM: = <.-MRTBL>/4

```
68 ;
69 ;
70 ;
71 ;
72 ;
73 000050 MRP::
74 000052 004767 000000G JSR PC,FIND ;LOCATE THE COMMAND IN THE COMMAND LINE
75 000054 103003 BCC 1$ ;OK, CONTINUE
76 000056 004767 000000G JSR PC,ERR4
77 000058 000422 BR MRPXX
78 000060 022700 1$: CMP #2,R0 ;COMMANDS ARE 2 CHARS
79 000062 001403 BEQ 2$
80 000064 004767 JSR PC,ERR8 ;INCORRECT CHAR COUNT
81 000066 000414 BR MRPXX ;TRY AGAIN
82 ;
83 000100 012700 000012 2$: MOV #MRNUM,R0 ;R0 = NUMBER OF COMMANDS
84 000102 012702 MOV #MRTBL,R2 ;R2 -> TABLE OF COMMAND MNEMONICS
85 000104 004767 JSR PC,SCAN ;FIND MATCH IN TABLE
86 000106 103003 BCC 3$ ;OK, CONTINUE
87 000108 004767 JSR PC,ERR12 ;COMMAND NOT IN TABLE
88 000110 000402 BR MRPXX ;TRY AGAIN
89 ;
90 ;
91 ;
92 000124 000171 000000 3$: JMP @ (R1)
93 ;
94 ;
95 ;
96 ;
97 ;
98 000130 LOCAL MRP LOOP. INCLUDE PROMPT FOR MRP COMMAND.
99 000132 012767 051115 000000G MRPXX:: MOV #MR.GCMBLK+G.DPRM+2 ;MOVE MRP NAME TO GCM BLOCK
100 000134 004767 JSR PC,GCONLY ;PROMPT
101 000136 004767 JSR PC,FIND ;LOCATE THE COMMAND IN THE COMMAND LINE
102 000138 103003 BCC 1$ ;OK, CONTINUE
103 000140 004767 JSR PC,ERR4
104 000142 000765 BR MRPXX
105 000144 022700 1$: CMP #2,R0 ;COMMANDS ARE 2 CHARS
106 000146 001403 BEQ 2$
107 000148 004767 JSR PC,ERR8 ;INCORRECT CHAR COUNT
108 000150 000757 BR MRPXX ;TRY AGAIN
109 ;
110 000172 012700 000012 2$: MOV #MRNUM,R0 ;R0 = NUMBER OF COMMANDS
111 000174 012702 MOV #MRTBL,R2 ;R2 -> TABLE OF COMMAND MNEMONICS
112 000176 004767 JSR PC,SCAN ;FIND MATCH IN TABLE
113 000178 103003 BCC 3$ ;OK, CONTINUE
114 000180 042767 BIC #BREAK,BASE ;CLEAR BREAKPOINT FLAG
115 000182 000167 JMP COMXX ;RETURN TO QMAIN. LOOK FOR PROCESSOR MNEMONIC
116 ;
117 ;
118 ;
119 000222 000171 000000 3$: JMP @ (R1)
120 ;
121 000001 .END
```

ALUCKE = 040002	BYTE4 = 000004	BYTE91 = 000133	MREN2 = 020000	Q\$NCLK = 176000
ALUOE = 004000	BYTE40 = 000050	BYTE92 = 000134	MRNUM = 000012	Q\$PP = 000100
AT1 = 000000 GX	BYTE41 = 000051	BYTE93 = 000135	MRP = 000050RG	Q\$PPSW = 000320
A01 = 010000	BYTE42 = 000052	BYTE94 = 000136	MRPXX = 000130RG	Q\$PP2 = 000300
BASE = 000000 GX	BYTE43 = 000053	BYTE95 = 000137	MRTBL = 000000RG	Q\$QHLT = 000013
B1TVAL = 000000	BYTE44 = 000054	BYTE96 = 000140	MSYN = 000040	Q\$QL = 000043
BIT0 = 000001	BYTE45 = 000055	BYTE97 = 000141	N = 000144	Q\$QLA = 000053
BIT1 = 000002	BYTE46 = 000056	BYTE98 = 000142	OF1 = 000000 GX	Q\$QLB = 000054
BIT10 = 002000	BYTE47 = 000057	BYTE99 = 000143	PLB = 000010	Q\$QLR = 000001
BIT11 = 004000	BYTE48 = 000058	BYTVAL = 000144	PLC = 000020	Q\$QW = 000042
BIT12 = 010000	BYTE49 = 000059	CBKALL = 001000	PLD = 000030	Q\$RDCD = 000005
BIT13 = 020000	BYTE5 = 000005	CBKCLK = 000400	PLRWR = 000200	Q\$RDMD = 000006
BIT14 = 040000	BYTE50 = 000062	CL1 = 000000 GX	PLR.EN = 000200	Q\$REBK = 001000
BIT15 = 100000	BYTE51 = 000063	CNOBRE = 100000	PR1 = 000000 GX	Q\$RNC = 006000
BIT2 = 000004	BYTE52 = 000064	COMXX = 000000 GX	Q\$CR1 = 176420	Q\$RSC = 004000
BIT3 = 000010	BYTE53 = 000065	CPCCEN = 010000	Q\$CR2 = 176422	Q\$RSET = 000010
BIT4 = 000020	BYTE54 = 000066	CPREAD = 040000	Q\$CLBR = 176424	Q\$SM = 100000
BIT5 = 000040	BYTE55 = 000067	CPWRITE = 020000	Q\$ATTN = 000100	Q\$SP = 000120
BIT6 = 000100	BYTE56 = 000070	CSABRD = 000004	Q\$BCL = 000001	Q\$SP2 = 000340
BIT7 = 000200	BYTE57 = 000071	CSEQCI = 100000	Q\$CCCP = 000040	RE1 = 000000 GX
BIT8 = 000400	BYTE58 = 000072	CSD = 000040	Q\$CHB = 000400	RGQ.EN = 000200
BIT9 = 001000	BYTE59 = 000073	CSWRITE = 000100	Q\$CHRL = 000200	RGQ.VA = 020000
BREAK = 000000 GX	BYTE6 = 000006	DBR.RD = 000001	Q\$CLR = 000040	RS1 = 000000 GX
BYTE0 = 000000	BYTE60 = 000074	DB\$CPP = 001457	Q\$CNC = 030000	SCAN = 000000 GX
BYTE1 = 000001	BYTE61 = 000075	DB\$SPT = 000026	Q\$CP = 000060	SEQ.CI = 000010
BYTE10 = 000012	BYTE62 = 000076	DB\$TPC = 000023	Q\$CPCC = 000010	SSI = 000000 GX
BYTE11 = 000013	BYTE63 = 000077	DISPGS = 100000	Q\$CP2 = 000260	ST1 = 000000 GX
BYTE12 = 000014	BYTE64 = 000100	DMAAUR = 000005	Q\$CSC = 010000	S\$CLR = 000000
BYTE13 = 000015	BYTE65 = 000101	DMARRD = 000003	Q\$CSEL = 000360	S\$LA = 000001
BYTE14 = 000016	BYTE66 = 000102	DMARWR = 000004	Q\$CSET = 000002	S\$QB = 000005
BYTE15 = 000017	BYTE67 = 000103	ENBR = 010000	Q\$CSP = 020000	S\$QR = 000006
BYTE16 = 000020	BYTE68 = 000104	ERR12 = 000000 GX	Q\$DMA = 000001	S\$QX = 000004
BYTE17 = 000021	BYTE69 = 000105	ERR4 = 000000 GX	Q\$ENBK = 040000	S\$SR = 000007
BYTE18 = 000022	BYTE7 = 000007	ERR8 = 000000 GX	Q\$ENOP = 020000	S\$S1 = 000010
BYTE19 = 000023	BYTE70 = 000106	FIND = 000000 GX	Q\$FAL = 004000	S\$S2 = 000014
BYTE2 = 000002	BYTE71 = 000107	GCMCLK = 000000 GX	Q\$FC = 000045	TD\$CTR = 176370
BYTE20 = 000024	BYTE72 = 000110	GCONLY = 000000 GX	Q\$FO = 000044	TD\$CTW = 176360
BYTE21 = 000025	BYTE73 = 000111	GO1 = 000000 GX	Q\$FP = 000046	TD\$INL = 004000
BYTE22 = 000026	BYTE74 = 000112	G.DPRM = 000000 GX	Q\$HBF = 000002	TD\$MEM = 000270
BYTE23 = 000027	BYTE75 = 000113	LD1 = 000000 GX	Q\$ICP = 000006	TD\$OAR = 176344
BYTE24 = 000030	BYTE76 = 000114	LOC.EN = 000100	Q\$IH = 000003	TD\$OTR = 176346
BYTE25 = 000031	BYTE77 = 000115	LOC.WA = 040000	Q\$IHRL = 000002	TD\$ORD = 000274
BYTE26 = 000032	BYTE78 = 000116	LOC.WB = 100000	Q\$IMRP = 000007	TD\$SW = 176376
BYTE27 = 000033	BYTE79 = 000117	MAREN1 = 000001	Q\$LBD = 001000	TD\$TAR = 176372
BYTE28 = 000034	BYTE8 = 000010	MAREN2 = 004000	Q\$LBDF = 001001	TD\$TAW = 176362
BYTE29 = 000035	BYTE80 = 000120	MARLOD = 010000	Q\$LBP = 000001	TD\$TDR = 176374
BYTE3 = 000003	BYTE81 = 000121	MAROUT = 000002	Q\$LCD = 000003	TD\$TDW = 176364
BYTE30 = 000036	BYTE82 = 000122	MAR.LO = 002000	Q\$LMD = 000004	T\$AD = 000020
BYTE31 = 000037	BYTE83 = 000123	MAR.OU = 000040	Q\$LDPP = 002000	T\$BA = 000002
BYTE32 = 000040	BYTE84 = 000124	MBKALL = 001000	Q\$LHP = 010000	T\$BD = 000010
BYTE33 = 000041	BYTE85 = 000125	MBKCLK = 000400	Q\$MNC = 140000	T\$BSO = 100000
BYTE34 = 000042	BYTE86 = 000126	MMABRD = 000100	Q\$MR = 000052	T\$BT = 000020
BYTE35 = 000043	BYTE87 = 000127	MMLEFT = 000002	Q\$MRP = 000040	T\$BTAR = 000030
BYTE36 = 000044	BYTE88 = 000130	MMOE = 000004	Q\$MRB = 000240	T\$BTOT = 002000
BYTE37 = 000045	BYTE89 = 000131	MMWRTE = 000010	Q\$MSC = 040000	T\$CD = 000100
BYTE38 = 000046	BYTE9 = 000011	MNOBRE = 100000	Q\$MSET = 000004	T\$CLK = 002000
BYTE39 = 000047	BYTE90 = 000132	MREN1 = 000001	Q\$MSP = 100000	T\$DISK = 000200

MRP- M00-M1110 27-MAR-80 15:11 PAGE 7-2
SYMBOL TABLE

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

T\$DRD = .000004	T\$UBUS = .100000	WORD3 = .000006	WORD54 = .000154	WORD79 = .000290
T\$EMEM = .010000	T\$1CLK = .000400	WORD30 = .000074	WORD55 = .000156	WORD8 = .000020
T\$FSAA = .000000	T\$8BEN = .000020	WORD31 = .000076	WORD56 = .000160	WORD80 = .000240
T\$FSAB = .000004	UBD, IN = .000020	WORD32 = .000100	WORD57 = .000162	WORD81 = .000242
T\$FSAC = .000014	WORD0 = .000000	WORD33 = .000102	WORD58 = .000164	WORD82 = .000244
T\$FSB2 = .000010	WORD1 = .000002	WORD34 = .000104	WORD59 = .000166	WORD83 = .000246
T\$IB = .000026	WORD10 = .000024	WORD35 = .000106	WORD6 = .000014	WORD84 = .000250
T\$IBAR = .000024	WORD11 = .000026	WORD36 = .000110	WORD60 = .000170	WORD85 = .000252
T\$IBE = .020000	WORD12 = .000030	WORD37 = .000112	WORD61 = .000172	WORD86 = .000254
T\$IBF = .040000	WORD13 = .000032	WORD38 = .000114	WORD62 = .000174	WORD87 = .000256
T\$ICD = .000040	WORD14 = .000034	WORD39 = .000116	WORD63 = .000176	WORD88 = .000260
T\$MODE = .004000	WORD15 = .000036	WORD4 = .000010	WORD64 = .000200	WORD89 = .000262
T\$OB = .000036	WORD16 = .000040	WORD40 = .000120	WORD65 = .000202	WORD9 = .000022
T\$OBE = .004000	WORD17 = .000042	WORD41 = .000122	WORD66 = .000204	WORD90 = .000264
T\$OBF = .010000	WORD18 = .000044	WORD42 = .000124	WORD67 = .000206	WORD91 = .000266
T\$OBRA = .000034	WORD19 = .000046	WORD43 = .000126	WORD68 = .000210	WORD92 = .000270
T\$OBWA = .000032	WORD2 = .000004	WORD44 = .000130	WORD69 = .000212	WORD93 = .000272
T\$OUTA = .100000	WORD20 = .000050	WORD45 = .000132	WORD7 = .000016	WORD94 = .000274
T\$RBD0 = .000200	WORD21 = .000052	WORD46 = .000134	WORD70 = .000214	WORD95 = .000276
T\$RNB = .000040	WORD22 = .000054	WORD47 = .000136	WORD71 = .000216	WORD96 = .000300
T\$RSET = .040000	WORD23 = .000056	WORD48 = .000140	WORD72 = .000220	WORD97 = .000302
T\$SC = .000022	WORD24 = .000060	WORD49 = .000142	WORD73 = .000222	WORD98 = .000304
T\$SCLK = .020000	WORD25 = .000062	WORDS = .000012	WORD74 = .000224	WORD99 = .000306
T\$SEG1 = .000000	WORD26 = .000064	WORD50 = .000144	WORD75 = .000226	WRDVAL = .000310
T\$SEG2 = .000001	WORD27 = .000066	WORD51 = .000146	WORD76 = .000230	XTREAD = .001000
T\$SEG3 = .000002	WORD28 = .000070	WORD52 = .000150	WORD77 = .000232	XTWRITE = .000400
T\$SO = .000001	WORD29 = .000072	WORD53 = .000152	WORD78 = .000234	

. ABS. 000000 000
000000 001
MRP. 000226 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3151 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:41
MRP,MRP,SP=[20,1]IM,[20,1]MRP

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
1 .TITLE--MRLD
2 000000 .PSECT: MRLD
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22 000000 LD1TBL:
23 000000 115 115 .ASCII: /MM/ ;LOAD MICROPGM MEMORY.
24 000002 000262 .WORD: LD1MM. ;LOAD DATA MEMORY (X AND T TABLES)
25 000004 115 104 .ASCII: /MD/
26 000006 000424 .WORD: LD1MD.
27 000002 LD1LN: == <.-LD1TBL>/4
28
29
30
31
32
33
34
35
36
37 000010 LD1:
38 000010 CALL: FIND ;LOCATE MEMORY MNEMONIC IN COMMAND LINE.
39 000014 103004 BCC: 1$ ;OK, CONTINUE.
40 000016 CALL: ERR4 ;MISSING OPERAND?
41 000022 000167 000644 JMP: LD1X ;EXIT.
42
43
44
45 000026 012700 000002 1$: MOV: #LD1LN,R0 ;NUMBER OF TABLE ENTRIES.
46 000032 012702 000000 MOV: #LD1TBL,R2 ;POINT TO TABLE.
47 000036 CALL: SCAN ;MATCH AGAINST COMMAND LINE.
48 000042 103004 BCC: 2$ ;MATCH WAS MADE.
49 000044 CALL: ERR7 ;INVALID MEMORY MNEMONIC?
50 000050 000167 000616 JMP: LD1X
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```


Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

58      ;      WILL TERMINATE THE LOAD.
59      ;      MR>LD MD 0
60      ;      2. START ADDRESS, LOOP INDICATOR, LOOP ON THE LOADING OF
61      ;      THIS ONE MEMORY LOCATION ONLY.
62      ;      MR>LD MD 0 L
63      ;      3. START ADDRESS, END ADDRESS, NO LOOP, ISSUE ONLY ONE
64      ;      PROMPT AND FILL MEMORY (BETWEEN START AND END ADDRESSES)
65      ;      WITH THIS VALUE.
66      ;      MR>LD MD 0 7
67      ;      4. START ADDRESS, END ADDRESS, LOOP INDICATOR, ISSUE ONLY ONE
68      ;      PROMPT AND LOOP ON THE LOADING OF MEMORY (BETWEEN START
69      ;      AND END ADDRESSES) WITH THIS VALUE.
70      ;      MR>LD MD 0 7 L
71      ;
72      ;
73 000054 010167 000000G 2$: MOV R1,RTNPT ;SAVE POINTER
74 000060      ;      CALL FIND ;LOCATE START ADDRESS IN COMMAND LINE
75 000064 103004      ;      BCC 3$ ;OK, CONTINUE
76 000066      ;
77 000072 000167 000574      ;      CALL ERR4 ;MISSING OPERAND
78 000076      ;      JMP LD1X ;EXIT
79 000102 103004      ;      3$: CALL PACK ;CONVERT COMMAND LINE VALUE TO BINARY
80 000104      ;      BCC 4$ ;CONVERSION SUCCESSFUL
81 000110 000167 000556      ;      CALL ERR5 ;INVALID NUMERIC VALUE
82 000112      ;      JMP LD1X
83 000114 016767 000000G-000000G 4$: MOV BINWD,MSTRT ;SAVE LOADING START ADDRESS
84 000122 016767 000000G-000000G      ;      MOV BINWD,MSTR2 ;SAVE IT TWICE (FOR REFRESH ON LOOP)
85 000130 012767 177777 000000G      ;      MOV #-1,MEND ;INIT END ADDRESS
86      ;
87      ;      START ADDRESS HAS BEEN FOUND, CAN FOR END ADDRESS OR LOOP
88      ;      INDICATOR (CONDITIONS 2-4 ABOVE). IF THERE IS NOTHING
89      ;      FURTHER IN THE COMMAND LINE, CONDITION 1 IS IN EFFECT.
90      ;
91 000136      ;      CALL FIND ;SCAN COMMAND LINE
92 000142 103004      ;      BCC 5$ ;SOMETHING THERE
93 000144 352767 000000G-000000G      ;      BIS #RP,BASE ;SIGNAL TO REPEAT PROMPT
94 000152 000437      ;      BR 9$ ;JUMP TO RTN
95      ;
96 000154 122711 000114      ;      5$: CMPB #L,(R1) ;LOOP INDICATOR
97 000160 001006      ;      BNE 6$ ;NO, MUST BE UPPER ADDRESS
98 000162 016767 000000G-000000G      ;      MOV MSTRT,MEND ;SET END ADDR = START ADDR
99 000170      ;      CALL HANG ;HOW TO STOP LOOP
100 000174 000426      ;      BR 9$ ;JUMP TO RTN
101      ;
102 000176      ;      6$: CALL PACK ;CONVERT UPPER ADDRESS
103 000202 103004      ;      BCC 7$ ;OK, CONTINUE
104 000204      ;      CALL ERR5 ;INVALID NUMERIC
105 000210 000167 000456      ;      JMP LD1X ;EXIT
106      ;
107      ;      SAVE END ADDRESS (BINARY)
108      ;      CHECK FOR LOOP INDICATOR AFTER END ADDRESS (CONDITION 4)
109      ;
110 000214 016767 000000G-000000G 7$: MOV BINWD,MEND ;SET UP END ADDRESS
111 000222      ;      CALL FIND ;CHECK FOR LOOP INDICATOR
112 000226 103411      ;      BCS 9$ ;NO LOOP
113 000230 122711 000114      ;      CMPB #L,(R1) ;CORRECT INDICATOR
114 000234 001404      ;      BEQ 8$ ;YES, CONTINUE

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

115 000236          CALL  ERR11          ;LOOP OPTION ERROR
116 000242 000167 000424      JMP  LD1X
117 000246          CALL  HANG          ;HOW TO STOP LOOP
118
119 000252 016701 000000G    9$  MOV  RTNPT,R1      ;POINT TO ROUTINE
120 000256 000171 000000      JMP  @R1          ;JUMP TO ROUTINE
121
122
123
124          LOAD MICROPGM MEMORY
125 000262          LD1MM:
126 000262 016746 000000G    MOV  MMHIGH,-(SP)      ;SUPPLY UPPER MEMORY LIMIT
127 000266 016746 000000G    MOV  MMLOW,-(SP)      ;LOWER LIMIT
128 000272          CALL  BUFS2          ;PREPARE TO LOAD
129 000276 103002          BCC  100$      ;NO ERRORS, CONTINUE
130 000300 000167 000366      JMP  LD1X
131
132 000304 032767 000000G-000000G 100$ BIT  #RP,BASE      ;REPEAT PROMPT
133 000312 001416          BEQ  2$          ;NO, ONCE ONLY
134 000314          1$  CALL  PDATA          ;READ DATA FROM COMMAND LINE
135 000320 103002          BCC  10$          ;NOT END OF MEMORY
136 000322 000167 000344      JMP  LD1X          ;IF END, EXIT
137 000326 102002          10$ BVC  20$          ;<CR> RESPONSE
138 000330 000167 000336      JMP  LD1X          ;YES, EXIT
139 000334          20$ CALL  LOADMM          ;LOAD MICRO MEMORY
140 000340 066767 000000G-000000G ADD  INCVAL,MSTR2      ;BUMP TO NEXT ADDRESS
141 000346 000762          BR  1$          ;REPEAT
142
143
144          PROMPT ONCE THEN FILL MEMORY
145 000350          2$  CALL  PDATA          ;PROMPT
146 000354 103002          BCC  30$          ;NOT END OF MEMORY OR ERROR
147 000356 000167 000310      JMP  LD1X          ;ELSE EXIT
148
149 000362          30$ CALL  LOADMM          ;LOAD MICRO MEMORY
150 000366 066767 000000G-000000G ADD  INCVAL,MSTR2      ;ADVANCE ADDRESS
151 000374 026767 000000G-000000G CMP  MSTR2,MEND      ;HAS UPPER MEMORY LIMIT BEEN REACHED
152 000402 101767          BLOS 30$          ;NO, CONTINUE
153 000404 032767 000000G-000000G BIT  #LOOP,BASE      ;LOOP ON
154 000412 001527          BEQ  LD1X          ;NO, EXIT
155 000414 016767 000000G-000000G MOV  MSTR,MSTR2      ;REINITIALIZE ADDRESS
156 000422 000757          BR  30$
157
158
159          LOAD DATA MEMORY
160
161 000424          LD1MD:
162 000424 016746 000000G    MOV  MDHIGH,-(SP)      ;SUPPLY UPPER MEMORY LIMIT
163 000430 016746 000000G    MOV  MDLOW,-(SP)      ;LOWER LIMIT
164 000434          CALL  BUFS2          ;PREPARE FOR LOAD
165 000440 103514          BCS  LD1X          ;ERROR
166
167 000442 012746 000377      MOV  #377,-(SP)      ;SET MRP MICRO ADDRESS = X'FF' (JUMP SELF)
168 000446          CALL  SEQMM
169
170 000452 005046          CLR  -(SP)          ;REINHIBIT BRANCH CONTROL REG
171 000454          CALL  MRPCR

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

172 000460 005046 CLR - (SP)
173 000462 CALL LBMRP ; SINGLE CLOCK TO REINHIBIT BRANCH REGISTER
174 ;
175 000466 005046 CLR - (SP) ; START MICROCODE AT 0
176 000470 CALL SEQCS
177 ;
178 000474 005046 CLR - (SP) ; REINHIBIT BRANCH CONTROL REG
179 000476 CALL CPR
180 000502 005046 CLR - (SP)
181 000504 CALL LBOP ; SINGLE CLOCK TO REINHIBIT BRANCH REGISTER
182 ;
183 000510 012767 001000 176422 MOV #Q$REBK,QR$CR2 ; RE-ARM INTERRUPTS
184 000516 012767 120000 176422 MOV #<Q$SM+Q$ENDP>,QR$CR2 ; SET SEARCH MODE + ENABLE INTERRUPTS
185 000524 012746 000360 MOV #Q$SEL,-(SP) ; CLEAR ALL SELECTIONS
186 000530 052716 001001 BIS #<Q$LBD+Q$LBP>,(SP) ; CLEAR DRIVE AND PULSE
187 000534 052716 170000 BIS #<Q$MNC+Q$CNC>,(SP) ; CLEAR CP NO-CLOCK
188 000540 005046 CLR - (SP) ; SET NOTHING
189 000542 CALL CSR1
190 ;
191 000546 032767 000000G-000000G BIT #RP,BASE ; REPEAT PROMPT
192 000554 001412 BEQ 2$ ; NO, ONCE ONLY
193 000556 1$: CALL PDATA ; READ DATA FROM COMMAND LINE
194 000562 103434 BCS LDMDX ; EXIT ON END OF MEMORY OR ERROR
195 000564 102433 BVS LDMDX ; <CR> RESPONSE TO PROMPT
196 000566 CALL LOADMD ; LOAD MRP DATA MEMORY
197 000572 066767 000000G-000000G ADD INCVAL,MSTR2 ; BUMP ADDRESS
198 000600 000766 BR 1$ ; REPEAT
199 ;
200 ; PROMPT ONCE THEN FILL MEMORY
201 ;
202 000602 2$: CALL PDATA ; PROMPT
203 000606 103422 BCS LDMDX ; END OF MEMORY OR ERROR
204 000610 102421 BVS LDMDX ; <CR> RESPONSE
205 000612 3$: CALL LOADMD ; LOAD DATA MEMORY
206 000616 066767 000000G-000000G ADD INCVAL,MSTR2 ; ADVANCE ADDRESS
207 000624 026767 000000G-000000G CMP MSTR2,MEND ; HAS UPPER MEMORY LIMIT BEEN REACHED
208 000632 101767 BLOS 3$ ; NO, CONTINUE
209 000634 032767 000000G-000000G BIT #LOOP,BASE ; LOOP ON
210 000642 001404 BEQ LDMDX ; EXIT
211 000644 016767 000000G-000000G MOV MSTRT,MSTR2 ; REINITIALIZE ADDRESS
212 000652 000757 BR 3$
213 ;
214 000654 005046 LDMDX: CLR - (SP) ; CLEAR NOTHING IN CSR1
215 000656 012746 176000 MOV #Q$NCLK,-(SP) ; SET NO-CLOCKS
216 000662 CALL CSR1
217 000666 005067 176422 CLR QR$CR2 ; SET LOAD MODE
218 ;
219 ;
220 000672 LD1X:
221 000672 042767 000000G-000000G BIC #RP,BASE ; CLEAR PROMPT FLAG
222 000700 CALL KILL ; KILL AST (IF THERE WAS ONE)
223 000704 000167 000000G JMP MRPIX

```

```

225      ;
226      ;
227      ;      LOAD:MRP:MICROPGM:MEMORY:
228      ;
229      ;
230 000710      LOADMM:
231 000710 016746 000000G      MOV.      MSTR2,-(SP)      ;SEND CURRENT ADDRESS.
232 000714      CALL.      SEQMM:      ;SEQUENCE UP TO START ADDR.
233 000720 012746 000012      MOV.      *(<MMJRTEN+MMLEFT>,-(SP)
234 000724      CALL.      MRPCRA:      ;DIRECT CNTL WORD TO MRP.
235 000730 016746 000000G      MOV.      DATA1,-(SP)      ;MOVE DATA WORD TO LOD BUS REG
236 000734      CALL.      LBMSC:      ;SEND DATA WORD TO MRP.      NS.
237      ;
238 000740 016746 000000G      MOV.      MSTR2,-(SP)      ;SEND CURRENT ADDRESS.
239 000744      CALL.      SEQMM:      ;SEQUENCE UP TO START ADDRESS.
240 000750 012746 000010      MOV.      *(<MMJRTEN>,-(SP)
241 000754      CALL.      MRPCRA:      ;DIRECT CNTL WORD TO MRP.
242 000760 016746 000000G      MOV.      DATA2,-(SP)      ;MOVE DATA WORD TO LOD BUS REG
243 000764      CALL.      LBMSC:      ;SEND DATA WORD TO MRP.      NS.
244      ;
245 000770 005046      CLR.      -(SP)
246 000772      CALL.      MRPCR:      ; CLEAR THE MRP CR BITS.      NS.
247 000776 005046      CLR.      -(SP)
248 001000      CALL.      LBMRP:      ;ISSUE CLOCK TO REINHIBIT BR.
249 001004      RETURN.

```

```

251
252
253
254
255
256 001006
257 001006 012767 000004 176424
258 001014 012767 120100 176422
259 001022 016701 176422
260 001026 032701 000100
261 001032 001373
262
263 001034 016767 000000G 176424
264 001042 012767 120040 176422
265 001050 016701 176422
266 001054 032701 000040
267 001060 001373
268
269 001062 012767 000001 176424
270 001070 012767 120040 176422
271 001076 016701 176422
272 001102 032701 000040
273 001106 001373
274
275 001110 012767 000000G 176424
276 001116 012767 120040 176422
277
278
279
280 001124
281
282 001136
283
284
285
286 001150 012767 100400 176422
287 001156 012767 101000 176422
288 001164 012767 160000 176422
289 001172
290
291 000001

```

LOAD-MRP-DATA-MEMORY.

LOADMD:

```

MOV. #Q$LDMD,QR$LBR ;MOVE-ATTN CODE-TO-LOD-BUS-REG
MOV. #<Q$ATTN+Q$SM+Q$ENOP>,QR$CR2 ;SET-ATTN CODE-READY
1$: MOV. QR$CR2,R1 ;READ-CSR2
BIT. #Q$ATTN,R1 ;ATTN-CLEAR
BNE. 1$ ;NO-READ-AGAIN
;
MOV. MSTR2,QR$LBR ;CD-MEMORY-START-ADDRESS-THRU-
MOV. #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET-CC-TO-CP
2$: MOV. QR$CR2,R1 ;READ-CSR2
BIT. #Q$CCCP,R1 ;IS-CC-TO-CP-CLEAR
BNE. 2$ ;NO-READ-AGAIN
;
MOV. #1,QR$LBR ;TRANSFER-COUNT--1-WORD
MOV. #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET-CC-TO-CP
3$: MOV. QR$CR2,R1 ;READ-CSR2
BIT. #Q$CCCP,R1 ;IS-CC-TO-CP-CLEAR
BNE. 3$ ;NO-READ-AGAIN
;
MOV. #DATA1,QR$LBR ;CC-MEMORY-DATA-BUFFER
MOV. #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET-CC-TO-CP
;
WAIT-FOR-INTERRUPT-FROM-CP.
WTSE$. #EFN.3
;
CLEF$. #EFN.3
;
RE-ARM-INTERRUPTS.
;
MOV. #<Q$SM+Q$CHB>,QR$CR2 ;CLEAR-INTERRUPT-(USE-HIT-BUFFER-INT)
MOV. #<Q$SM+Q$REBK>,QR$CR2 ;RE-ARM
MOV. #<Q$SM+Q$ENBK+Q$ENOP>,QR$CR2 ;ENABLE
RETURN.
;
.END.

```

ALUCKE = 040000	BYTE39 = 000047	BYTE90 = 000132	LOC.WB = 100000	Q\$FC = 000045
ALUOE = 004000	BYTE4 = 000004	BYTE91 = 000133	LOOP = ***** GX	Q\$FO = 000044
A01 = 010000	BYTE40 = 000050	BYTE92 = 000134	MAREN1 = 000001	Q\$FP = 000046
BASE = ***** GX	BYTE41 = 000051	BYTE93 = 000135	MAREN2 = 004000	Q\$HBF = 000002
BINWD = ***** GX	BYTE42 = 000052	BYTE94 = 000136	MARLOD = 010000	Q\$ICP = 000006
BITVAL = 000000	BYTE43 = 000053	BYTE95 = 000137	MAROUT = 000002	Q\$IFB = 000003
BIT0 = 000001	BYTE44 = 000054	BYTE96 = 000140	MAR.LO = 002000	Q\$IHLR = 000002
BIT1 = 000002	BYTE45 = 000055	BYTE97 = 000141	MAR.OU = 000040	Q\$IMRP = 000007
BIT10 = 002000	BYTE46 = 000056	BYTE98 = 000142	MBKALL = 001000	Q\$LBD = 001000
BIT11 = 004000	BYTE47 = 000057	BYTE99 = 000143	MBKCLK = 000400	Q\$LBDF = 001001
BIT12 = 010000	BYTE48 = 000060	BYTVAL = 000144	MDHIGH = ***** GX	Q\$LBPF = 000001
BIT13 = 020000	BYTE49 = 000061	CBKALL = 001000	MDLOW = ***** GX	Q\$LCD = 000003
BIT14 = 040000	BYTE5 = 000005	CBKCLK = 000400	MEND = ***** GX	Q\$LDMD = 000004
BIT15 = 100000	BYTE50 = 000062	CNOBRE = 100000	MMADDR = 000100	Q\$LDPP = 002000
BIT2 = 000004	BYTE51 = 000063	CPCCEN = 010000	MMHIGH = ***** GX	Q\$LHP = 010000
BIT3 = 000010	BYTE52 = 000064	CPCR = ***** GX	MMLEFT = 000002	Q\$MNC = 140000
BIT4 = 000020	BYTE53 = 000065	CPREAD = 040000	MMLOW = ***** GX	Q\$MR = 000052
BIT5 = 000040	BYTE54 = 000066	CPWRTE = 020000	MMOE = 000004	Q\$MRP = 000040
BIT6 = 000100	BYTE55 = 000067	CSADRD = 000004	MMWRTE = 000010	Q\$MRP2 = 000240
BIT7 = 000200	BYTE56 = 000070	CSEOC = 100000	MNOBRE = 100000	Q\$MSC = 040000
BIT8 = 000400	BYTE57 = 000071	CSOE = 000040	MREN1 = 000001	Q\$MSET = 000004
BIT9 = 001000	BYTE58 = 000072	CSR1 = ***** GX	MREN2 = 020000	Q\$MSP = 100000
BUFSET = ***** GX	BYTE59 = 000073	CSWRTE = 000100	MRPCR = ***** GX	Q\$NCLK = 176000
BUFS2 = ***** GX	BYTE6 = 000006	DATA1 = ***** GX	MRPCRA = ***** GX	Q\$PP = 000100
BYTE0 = 000000	BYTE60 = 000074	DATA2 = ***** GX	MRPXY = ***** GX	Q\$PPSW = 000320
BYTE1 = 000001	BYTE61 = 000075	DBR.RD = 000001	MSTRT = ***** GX	Q\$PP2 = 000300
BYTE10 = 000012	BYTE62 = 000076	DB\$CPP = 001457	MSTR2 = ***** GX	Q\$QHLT = 000013
BYTE11 = 000013	BYTE63 = 000077	DB\$SPT = 000026	MSYN = 000040	Q\$QL = 000043
BYTE12 = 000014	BYTE64 = 000100	DB\$TPC = 000023	N = 000144	Q\$QLA = 000053
BYTE13 = 000015	BYTE65 = 000101	DISPGS = 100000	PACK = ***** GX	Q\$QLB = 000054
BYTE14 = 000016	BYTE66 = 000102	DMARWR = 000005	PDATA = ***** GX	Q\$QLR = 000001
BYTE15 = 000017	BYTE67 = 000103	DMARRD = 000003	PLB = 000010	Q\$QW = 000042
BYTE16 = 000020	BYTE68 = 000104	DMARWR = 000004	PLC = 000020	Q\$RDCD = 000005
BYTE17 = 000021	BYTE69 = 000105	EFN.3 = ***** GX	PLD = 000030	Q\$RDMD = 000006
BYTE18 = 000022	BYTE7 = 000007	ENBR = 010000	PLRWR = 000200	Q\$REBK = 001000
BYTE19 = 000023	BYTE70 = 000106	ERR11 = ***** GX	PLR.EN = 000200	Q\$RNC = 006000
BYTE2 = 000002	BYTE71 = 000107	ERR4 = ***** GX	QR\$CR1 = 176420	Q\$RSC = 004000
BYTE20 = 000024	BYTE72 = 000110	ERR5 = ***** GX	QR\$CR2 = 176422	Q\$RSET = 000010
BYTE21 = 000025	BYTE73 = 000111	ERR7 = ***** GX	Q\$CLBR = 176424	Q\$SM = 100000
BYTE22 = 000026	BYTE74 = 000112	FIND = ***** GX	Q\$ATTN = 000100	Q\$SP = 000120
BYTE23 = 000027	BYTE75 = 000113	HANG = ***** GX	Q\$BCL = 000001	Q\$SP2 = 000340
BYTE24 = 000030	BYTE76 = 000114	INCVAL = ***** GX	Q\$CCCP = 000040	RGQ.EN = 000200
BYTE25 = 000031	BYTE77 = 000115	KILL = ***** GX	Q\$CHB = 000400	RGQ.VA = 020000
BYTE26 = 000032	BYTE78 = 000116	LBCP = ***** GX	Q\$CHRL = 000200	RP = ***** GX
BYTE27 = 000033	BYTE79 = 000117	LBMRP = ***** GX	Q\$CLR = 000040	RTNPT = ***** GX
BYTE28 = 000034	BYTE8 = 000010	LBMSC = ***** GX	Q\$CNC = 030000	SCAN = ***** GX
BYTE29 = 000035	BYTE80 = 000120	LDMDX = 000654R	002.Q\$CP = 000060	SEQCS = ***** GX
BYTE3 = 000003	BYTE81 = 000121	LD1 = 000010RG	002.Q\$CPCC = 000010	SEQMM = ***** GX
BYTE30 = 000036	BYTE82 = 000122	LD1LN = 000002.G	Q\$CP2 = 000260	SEQ.CI = 000010
BYTE31 = 000037	BYTE83 = 000123	LD1MD = 000424RG	002.Q\$CSC = 010000	S\$CLR = 000000
BYTE32 = 000040	BYTE84 = 000124	LD1MM = 000262RG	002.Q\$CSEL = 000360	S\$LA = 000001
BYTE33 = 000041	BYTE85 = 000125	LD1TBL = 000000RG	002.Q\$CSET = 000002	S\$OB = 000005
BYTE34 = 000042	BYTE86 = 000126	LD1X = 000672R	002.Q\$CSP = 020000	S\$OR = 000006
BYTE35 = 000043	BYTE87 = 000127	LOADMD = 001006R	002.Q\$DMA = 000001	S\$OX = 000004
BYTE36 = 000044	BYTE88 = 000130	LOADMM = 000710R	002.Q\$ENBK = 040000	S\$SR = 000007
BYTE37 = 000045	BYTE89 = 000131	LOC.EN = 000100	Q\$ENOP = 020000	S\$S1 = 000010
BYTE38 = 000046	BYTE9 = 000011	LOC.WB = 040000	Q\$FC = 000000	S\$S2 = 000014

MRLD... MACRO.M1110 27-MAR-80 15:10 PAGE 7-2
 SYMBOL TABLE

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

TD\$CTR= 176370	T\$ICD= 000040	WORD18= 000044	WORD46= 000134	WORD74= 000224
TD\$CTW= 176360	T\$MODE= 004000	WORD19= 000046	WORD47= 000136	WORD75= 000226
TD\$INL= 004000	T\$OB= 000036	WORD2= 000004	WORD48= 000140	WORD76= 000230
TD\$MEM= 000270	T\$OBE= 004000	WORD20= 000050	WORD49= 000142	WORD77= 000232
TD\$OAR= 176344	T\$OBF= 010000	WORD21= 000052	WORD5= 000012	WORD78= 000234
TD\$OTR= 176346	T\$OBRA= 000034	WORD22= 000054	WORD50= 000144	WORD79= 000236
TD\$QRD= 000274	T\$OBWA= 000032	WORD23= 000056	WORD51= 000146	WORD8= 000020
TD\$SW= 176376	T\$OUTA= 100000	WORD24= 000060	WORD52= 000150	WORD80= 000240
TD\$TAR= 176372	T\$RBD0= 000200	WORD25= 000062	WORD53= 000152	WORD81= 000242
TD\$TAU= 176362	T\$RNB= 000040	WORD26= 000064	WORD54= 000154	WORD82= 000244
TD\$TDR= 176374	T\$RSET= 040000	WORD27= 000066	WORD55= 000156	WORD83= 000246
TD\$TDW= 176364	T\$SC= 000022	WORD28= 000070	WORD56= 000160	WORD84= 000250
T\$AD= 000020	T\$SCLK= 020000	WORD29= 000072	WORD57= 000162	WORD85= 000252
T\$BA= 000002	T\$SEG1= 000000	WORD3= 000006	WORD58= 000164	WORD86= 000254
T\$BD= 000010	T\$SEG2= 000001	WORD30= 000074	WORD59= 000166	WORD87= 000256
T\$BSO= 100000	T\$SEG3= 000002	WORD31= 000076	WORD6= 000014	WORD88= 000260
T\$BT= 000020	T\$SO= 000001	WORD32= 000100	WORD60= 000170	WORD89= 000262
T\$BTAR= 000030	T\$UBUS= 100000	WORD33= 000102	WORD61= 000172	WORD9= 000022
T\$BTD= 002000	T\$1CLK= 000400	WORD34= 000104	WORD62= 000174	WORD90= 000264
T\$CD= 000100	T\$OBEN= 000020	WORD35= 000106	WORD63= 000176	WORD91= 000266
T\$CLK= 002000	UBD.IN= 000020	WORD36= 000110	WORD64= 000200	WORD92= 000270
T\$DISK= 000200	WORD0= 000000	WORD37= 000112	WORD65= 000202	WORD93= 000272
T\$DRD= 000004	WORD1= 000002	WORD38= 000114	WORD66= 000204	WORD94= 000274
T\$EMEM= 010000	WORD10= 000024	WORD39= 000116	WORD67= 000206	WORD95= 000276
T\$FSAA= 000000	WORD11= 000026	WORD4= 000010	WORD68= 000210	WORD96= 000300
T\$FSAB= 000004	WORD12= 000030	WORD40= 000120	WORD69= 000212	WORD97= 000302
T\$FSAC= 000014	WORD13= 000032	WORD41= 000122	WORD7= 000016	WORD98= 000304
T\$FSB2= 000010	WORD14= 000034	WORD42= 000124	WORD70= 000214	WORD99= 000306
T\$IB= 000026	WORD15= 000036	WORD43= 000126	WORD71= 000216	WRDVAL= 000310
T\$IBAR= 000024	WORD16= 000040	WORD44= 000130	WORD72= 000220	XTREAD= 001000
T\$IBE= 020000	WORD17= 000042	WORD45= 000132	WORD73= 000222	XTWTE= 000400
T\$IBF= 040000				

. ABS. 000000 000
 000000 001
 MRLD. 001174 002
 ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3608 WORDS (15 PAGES)
 DYNAMIC MEMORY: 4916 WORDS (18 PAGES)
 ELAPSED TIME: 00:00:48
 MRLD, MRLD /-SP=[20,1]IM,[20,1]MRLD

```

1          .TITLE MRPR
2 000000    .PSECT MRPR
3          ;
4          ;
5          ;
6          ;
7          ;
8          ;
9          ;
10         ;
11         ;
12         ;
13         ;
14         ;
15         ;
16         ;
17         ;
18         ;
19         ;
20         ;
21         ;
22         ;
23         ;
24         ;
25         ;
26         ;
27         ;
28         ;
29         ;
30         ;
31         ;
32 000000    PR1TBL:
33 000000      115      115      .ASCII /MM/          ;PRINT MICROPGM MEMORY
34 000002      000276    .WORD PRIMM
35 000004      115      104    .ASCII /MD/          ;PRINT DATA MEMORY
36 000006      000476    .WORD PRIMD
37 000002      000002    PR1LN: == <.-PR1TBL>4
38          ;
39          ;
40          ;
41          ;
42          ;
43          ;
44          ;
45          ;
46          ;
47 000010    PR1::
48 000010      103004    CALL FIND          ;LOCATE MEMORY MNEMONIC IN COMMAND LINE
49 000014      000002    BCC 1$            ;OK, CONTINUE
50 000016      000167    CALL ERR4          ;MISSING OPERAND
51 000022      000167    JMP PR1X          ;EXIT
52          ;
53 000026      012700    000002    1$: MOV #PR1LN,R0      ;NUMBER OF TABLE ENTRIES
54 000032      012702    000000    MOV #PR1TBL,R2      ;POINT TO TABLE
55 000036      103004    CALL SCAN          ;MATCH AGAINST COMMAND LINE
56 000042      000004    BCC 2$            ;MATCH WAS MADE
57 000044      000004    CALL ERR7          ;INVALID MEMORY MNEMONIC

```



```

58 000050 000167 001014 JMP .....PR1X
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78 000054 010167 000000G 2$: MOV R1,R1NPT. ;SAVE POINTER.
79 000060 CALL FIND ;LOCATE NON-BLANK IN COMMAND LINE.
80 000064 103004 BCC 3$ ;OK, CONTINUE.
81 000066 CALL ERR4 ;MISSING OPERAND.
82 000072 000167 000772 JMP PR1X ;EXIT.
83 000076 3$: CALL PACK ;CONVERT COMMAND LINE VALUE TO BINARY.
84 000102 103004 BCC 4$ ;CONVERSION SUCCESSFUL.
85 000104 CALL ERR5 ;INVALID NUMERIC VALUE.
86 000110 000167 000754 JMP PR1X
87
88 000114 016767 000000G 000000G 4$: MOV BINWD,MSTRT. ;SAVE PRINT START ADDRESS.
89 000122 016767 000000G 000000G MOV BINWD,MSTR2. ;SAVE IT TWICE.
90 000130 012767 177777 000000G MOV #-1,MEND. ;INIT END ADDRESS.
91
92
93
94
95
96 000136
97 000142 103004 CALL FIND ;SCAN COMMAND LINE.
98 000144 016767 000000G 000000G BCC 5$ ;SOMETHING THERE.
99 000152 000445 MOV MSTRT,MEND. ;SET END ADDR = START ADDR.
100 BR 9$ ;JUMP TO RTN.
101 000154 122711 000114 5$: CMPB #1L,(R1) ;LOOP INDICATOR.
102 000160 001011 BNE 6$ ;NO, MUST BE UPPER ADDRESS.
103 000162 052767 000000G 000000G BIS #OUT,BASE. ;SET FLAG FOR OUTPUT CONTROL.
104 000170 016767 000000G 000000G MOV MSTRT,MEND. ;SET END ADDR = START ADDR.
105 000176 CALL HANG ;HOW TO STOP LOOP.
106 000202 000431 BR 9$ ;JUMP TO RTN.
107
108 000204 6$: CALL PACK ;CONVERT UPPER ADDRESS.
109 000210 103004 BCC 7$ ;OK, CONTINUE.
110 000212 CALL ERR5 ;INVALID NUMERIC.
111 000216 000167 000646 JMP PR1X ;EXIT.
112
113
114

```

SAVE END ADDRESS (BINARY)
CHECK FOR LOOP INDICATOR AFTER END ADDRESS (CONDITION 4)

```
115
116 000222 016767 000000G-000000G-7$: MOV BINWD,MEND ;SET-UP-END-ADDRESS.
117 000230 CALL FIND ;CHECK-FOR-LOOP-INDICATOR.
118 000234 103414 BCS 9$ ;NO-LOOP.
119 000236 122711 000114 CMPB #*L,(R1) ;CORRECT-LOOP-INDICATOR.
120 000242 001404 BEQ 8$ ;YES.
121 000244 CALL ERR11
122 000250 000167 000614 JMP PR1X ;NO.
123 000254 052767 000000G-000000G-8$: BIS #*OUT,BASE ;SET-OUTPUT-CONTROL.
124 000262 CALL HANG ;HOW-TO-STOP-LOOP.
125
126 000266 016701 000000G 9$: MOV RTNPT,R1 ;POINT-TO-ROUTINE.
127 000272 000171 000000 JMF 0(R1) ;JUMP-TO-ROUTINE.
128
129
130
131 PRINT-FROM-MICROPGM-MEMORY.
132 000276 PRIMM:
133 000276 016746 000000G MOV MMHIGH,-(SP) ;SUPPLY-UPPER-MEMORY-LIMIT.
134 000302 016746 000000G MOV MMLW,-(SP) ;LOWER-LIMIT.
135 000306 CALL BUFS2 ;PREPARE-FOR-LOAD.
136 000312 103002 BCC 1$ ;NO-ERROR.
137 000314 000167 000550 JMP PR1X ;ERROR.
138
139 000320 016746 000000G 1$: MOV MSTR2,-(SP) ;SEND-CURRENT-ADDRESS.
140 000324 CALL SEQMM ;SEQUENCE-UP-TO-PRINT-START-ADDRESS.
141 000330 012746 000006 MOV #<MMOE+MMLEFT>,-(SP)
142 000334 CALL MRPCR ;DIRECT-CNTL-WORD-TO-MRP.
143 000340 005046 CLR -(SP)
144 000342 CALL LBMRP ;ISSUE-CLOCK-TO-REINHIBIT-BR-SELECT.
145 000346 CALL MRPLB ;REQUEST-MRP-TO-LOD-BUS NS.
146 000352 012667 000000G MOV (SP)+,DATA1 ;GET-MRP-WORD-FROM-STACK.
147
148 000356 016746 000000G MOV MSTR2,-(SP) ;SEND-CURRENT-ADDRESS.
149 000362 CALL SEQMM ;SEQUENCE-UP-TO-PRINT-START-ADDRESS.
150 000366 012746 000004 MOV #<MMOE>,-(SP)
151 000372 CALL MRPCR ;SELECT-RIGHT-HALF-OF-PLR.
152 000376 005046 CLR -(SP) ;DIRECT-CNTL-WORD-TO-MRP.
153 000400 CALL LBMRP ;CLOCK-TO-REINHIBIT-BRANCH-REGISTER.
154 000404 CALL MRPLB ;REQUEST-MRP-TO-LOD-BUS NS.
155 000410 012667 000000G MOV (SP)+,DATA2 ;GET-MRP-WORD-FROM-STACK.
156 000414 005046 CLR -(SP) ;CLEAR-MRP-CONTROL-REG.
157 000416 CALL MRPCR
158
159 000422 CALL PRDATA ;PRINT-MEMORY-CONTENTS.
160 000426 103002 BCC 2$ ;NOT-END-OF-MEMORY.
161 000430 000167 000434 JMP PR1X ;END-OF-MEMORY.
162
163 000434 065767 000000G-000000G-2$: ADD INCVAL,MSTR2 ;ADVANCE-ADDRESS.
164 000442 026767 000000G-000000G CMP MSTR2,MEND ;HAS-UPPER-MEMORY-LIMIT-BEEN-REACHED.
165 000450 101723 BLOS 1$ ;NO-CONTINUE.
166 000452 032767 000000G-000000G BIT #LOOP,BASE ;LOOP-ON-?
167 000460 001002 BNE 3$ ;YES-CONTINUE.
168 000462 000167 000402 JMP PR1X ;EXIT.
169 000466 016767 000000G-000000G-3$: MOV MSTR,MSTR2 ;AND-START-ADDRESS.
170 000474 000711 BR 1$ ;AND-REPEAT-PRINT.
171
```

```
172.
173.
174.
175 000476
176 000476 016746 000000G.
177 000502 016746 000000G.
178 000506
179 000512 103566
180
181 000514 012746 000377
182 000520
183
184 000524 005046
185 000526
186 000532 005046
187 000534
188
189 000540 005046
190 000542
191
192 000546 005046
193 000550
194 000554 005046
195 000556
196
197 000562 012767 001000 176422.
198 000570 012767 120000 176422.
199 000576 012746 000360
200 000602 052716 001001
201 000606 052716 170000
202 000612 005046
203 000614
204
205 000620 012767 000006 176424
206 000626 012767 120100 176422.
207 000634 016701 176422
208 000640 032701 000100
209 000644 001373
210
211 000646 016767 000000G. 176424
212 000654 012767 120040 176422.
213 000662 016701 176422
214 000666 032701 000040
215 000672 001373
216
217 000674 012767 000001 176424
218 000702 012767 120040 176422.
219 000710 016701 176422
220 000714 032701 000040
221 000720 001373
222
223 000722 012767 000000G. 176424
224 000730 012767 120040 176422.
225
226
227
228 000736
```

PRINT FROM DATA MEMORY

PR1MD::

MOV. MDHIGH, -(SP) ; SUPPLY MEMORY UPPER LIMIT.

MOV. MDLOW, -(SP) ; LOWER LIMIT.

CALL. BUFSET. ; PREPARE FOR LOAD.

BCS. PR1X ; ERROR.

MOV. #377, -(SP) ; SET MRP MICRO ADDRESS = 'X'FF' (JUMP SELF)

CALL. SEQMM.

CLR. -(SP) ; REINHIBIT BRANCH CONTROL REG.

CALL. MRPCR.

CLR. -(SP)

CALL. LBMRP. ; SINGLE CLOCK TO REINHIBIT BRANCH REGISTER.

CLR. -(SP)

CALL. SEQCS. ; START MICROCODE AT 0

CLR. -(SP)

CALL. CPCR.

CLR. -(SP)

CALL. LBPC. ; SINGLE CLOCK TO REINHIBIT BRANCH REGISTER.

MOV. #Q\$REBK, QR\$CR2 ; RE-ARM INTERRUPTS.

MOV. #<Q\$SM+Q\$ENOP>, QR\$CR2. ; SET SEARCH MODE + ENABLE INTERRUPTS.

MOV. #Q\$CSEL, -(SP)

BIS. #<Q\$LBD+Q\$LBP>, (SP) ; CLEAR ALL SELECTIONS.

BIS. #<Q\$MNC+Q\$CNC>, (SP) ; CLEAR DRIVE AND PULSE.

CLR. -(SP) ; CLEAR CP NO-CLOCK.

CALL. CSR1 ; SET NOTHING.

PRMD: MOV. #Q\$RDMD, QR\$LBR ; MOVE ATTN CODE TO LOD BUS REG

MOV. #<Q\$ATTN+Q\$SM+Q\$ENOP>, QR\$CR2. ; SET ATTN CODE READY.

1\$: MOV. QR\$CR2, R1 ; READ CSR2

BIT. #Q\$ATTN, R1 ; ATTN CLEAR.

BNE. 1\$; NO, READ AGAIN.

MOV. MSTR2, QR\$LBR. ; CD MEMORY START ADDRESS.

MOV. #<Q\$CCCP+Q\$SM+Q\$ENOP>, QR\$CR2. ; SET CC TO CP.

2\$: MOV. QR\$CR2, R1 ; READ CSR2

BIT. #Q\$CCCP, R1 ; IS CC TO CP CLEAR.

BNE. 2\$; NO, READ AGAIN.

MOV. #1, QR\$LBR. ; TRANSFER COUNT = 1 WORD.

MOV. #<Q\$CCCP+Q\$SM+Q\$ENOP>, QR\$CR2. ; SET CC TO CP.

3\$: MOV. QR\$CR2, R1 ; READ CSR2

BIT. #Q\$CCCP, R1 ; IS CC TO CP CLEAR.

BNE. 3\$; NO, READ AGAIN.

MOV. #DATA1, QR\$LBR. ; CC MEMORY DATA BUFFER.

MOV. #<Q\$CCCP+Q\$SM+Q\$ENOP>, QR\$CR2. ; SET CC TO CP.

WAIT FOR INTERRUPT FROM CP.

WTSE\$. #EFN. 3

```
229      ;
230 000750      CLEF$S- #EFN.3
231      ;
232      ;
233      ;
234 000762- 012767 100400 176422-  MOV-  #<Q$SM+Q$CHB>,&CR2-  ;CLEAR- INTERRUPT- (USE- HIT- BUFFER- INT)
235 000770- 012767 101000 176422-  MOV-  #<Q$SM+Q$REBK>,&CR2-  ;RE-ARM-
236 000776- 012767 160000 176422-  MOV-  #<Q$SM+Q$ENBK+Q$ENOP>,&CR2- ;ENABLE-
237      ;
238 001004      CALL-  PRDATA-  ;PRINT- MEMORY- CONTENTS-
239 001010 103417      BCS-  PRMDX-  ;END- OF- MEMORY, EXIT
240      ;
241 001012- 066767 000000G-000000G-  ADD-  INCVAL,MSTR2-  ;ADVANCE- ADDRESS-
242 001020- 026767 000000G-000000G-  CMP-  MSTR2,MEND-  ;HAS- UPPER- MEMORY- LIMIT- BEEN- REACHED-
243 001026- 101674      BLOS-  PRMD  ;NO- CONTINUE-
244 001030- 032767 000000G-000000G-  BIT-  #LOOP,BASE-  ;LOOP- ON- ?
245 001036- 001404      BEQ-  PRMDX-  ;NO- EXIT-
246 001040- 016767 000000G-000000G-  MOV-  MSTR2,MSTR2-  ;INIT- START- ADDRESS-
247 001046- 000664      BR-  PRMD  ;AND- REPEAT- PRINT-
248      ;
249 001050- 005046      PRMDX: CLR-  -(SP)  ;CLEAR- NOTHING- IN- CSR1
250 001052- 012746 176000      MOV-  #Q$NCLK,-(SP)  ;SET- NO- CLOCKS-
251 001056      CALL-  CSR1
252 001062- 012767 000000 176422-  MOV-  #0,&CR2-  ;SET- LOAD- MODE-
253      ;
254      ;
255 001070      PR1X:
256 001076- 042767 000000C-000000G-  BIC-  #<ONCE+OUT>,&BASE-  ;CLEAR- PRINT- CONTROL- FLAGS-
257 001076      CALL-  KILL  ;KILL- AST- (IF- THERE- WAS- ONE)
258 001102- 300167 000000G-  JMP-  MRPXX-
259      ;
260 000001      .END-
```

ALUCKE = 040000	BYTE39 = 000047	BYTE90 = 000132	MDHIGH = ***** GX	Q\$FC = 000045
ALUOE = 004000	BYTE4 = 000004	BYTE91 = 000133	MDLOW = ***** GX	Q\$FO = 000044
A01 = 010000	BYTE40 = 000050	BYTE92 = 000134	MEND = ***** GX	Q\$FP = 000046
BASE = ***** GX	BYTE41 = 000051	BYTE93 = 000135	MMADRD = 000100	Q\$HBF = 000002
BINWD = ***** GX	BYTE42 = 000052	BYTE94 = 000136	MMHIGH = ***** GX	Q\$ICP = 000006
BITVAL = 000000	BYTE43 = 000053	BYTE95 = 000137	MMLEFT = 000002	Q\$IHB = 000003
BIT0 = 000001	BYTE44 = 000054	BYTE96 = 000140	MMLOW = ***** GX	Q\$IHRL = 000002
BIT1 = 000002	BYTE45 = 000055	BYTE97 = 000141	M10E = 000004	Q\$IMRP = 000007
BIT10 = 002000	BYTE46 = 000056	BYTE98 = 000142	MMWRTE = 000010	Q\$LBD = 001000
BIT11 = 004000	BYTE47 = 000057	BYTE99 = 000143	MNOBRE = 100000	Q\$LBDP = 001001
BIT12 = 010000	BYTE48 = 000060	BYTVAL = 000144	MREN1 = 000001	Q\$LBP = 000001
BIT13 = 020000	BYTE49 = 000061	CBKALL = 001000	MREN2 = 020000	Q\$LCD = 000003
BIT14 = 040000	BYTE5 = 000005	CBKCLK = 000400	MRPCR = ***** GX	Q\$LDMD = 000004
BIT15 = 100000	BYTE50 = 000062	CNOBRE = 100000	MRPLB = ***** GX	Q\$LDPP = 002000
BIT2 = 000004	BYTE51 = 000063	CPCCEN = 010000	MRPXX = ***** GX	Q\$LHP = 010000
BIT3 = 000010	BYTE52 = 000064	CPCR = ***** GX	MSTRT = ***** GX	Q\$MNC = 140000
BIT4 = 000020	BYTE53 = 000065	CPREAD = 040000	MSTR2 = ***** GX	Q\$MR = 000052
BIT5 = 000040	BYTE54 = 000066	CPWRTE = 020000	MSYN = 000040	Q\$MRP = 000040
BIT6 = 000100	BYTE55 = 000067	CSADRD = 000004	N = 000144	Q\$MRP2 = 000240
BIT7 = 000200	BYTE56 = 000070	CSEOCI = 100000	ONCE = ***** GX	Q\$MSC = 040000
BIT8 = 000400	BYTE57 = 000071	CSOE = 000040	OUT = ***** GX	Q\$MSET = 000004
BIT9 = 001000	BYTE58 = 000072	CSR1 = ***** GX	PACK = ***** GX	Q\$MSP = 100000
BUFSET = ***** GX	BYTE59 = 000073	CSWRTE = 000100	PLB = 000010	Q\$NCLK = 176000
BUF52 = ***** GX	BYTE6 = 000006	DATA1 = ***** GX	PLC = 000020	Q\$PP = 000100
BYTE0 = 000000	BYTE60 = 000074	DATA2 = ***** GX	PLD = 000030	Q\$PPSW = 000320
BYTE1 = 000001	BYTE61 = 000075	DBR:RD = 000001	PLRW = 000200	Q\$PP2 = 000300
BYTE10 = 000012	BYTE62 = 000076	DB\$CPP = 001457	PLR:EN = 000200	Q\$QHLT = 000013
BYTE11 = 000013	BYTE63 = 000077	DB\$SPT = 000026	PRDATA = ***** GX	Q\$QL = 000043
BYTE12 = 000014	BYTE64 = 000100	DB\$TPC = 000023	PRMD = 000620R	002 Q\$QLA = 000053
BYTE13 = 000015	BYTE65 = 000101	DISPGS = 100000	PRMDX = 001050R	002 Q\$QLB = 000054
BYTE14 = 000016	BYTE66 = 000102	DMHWR = 000005	PR1 = 000010RG	002 Q\$QLR = 000001
BYTE15 = 000017	BYTE67 = 000103	DMARRD = 000003	PRILN = 000002 G	Q\$QW = 000042
BYTE16 = 000020	BYTE68 = 000104	DMARWR = 000004	PRIND = 000476RG	002 Q\$RDCD = 000005
BYTE17 = 000021	BYTE69 = 000105	EFN:3 = ***** GX	PRIMM = 000276RG	002 Q\$RDMK = 000006
BYTE18 = 000022	BYTE7 = 000007	ENBR = 010000	PRITBL = 000000RG	002 Q\$REBK = 001000
BYTE19 = 000023	BYTE70 = 000106	ERR11 = ***** GX	PRIX = 001070R	002 Q\$RNC = 006000
BYTE2 = 000002	BYTE71 = 000107	ERR4 = ***** GX	Q\$CR1 = 176420	Q\$RSC = 004000
BYTE20 = 000024	BYTE72 = 000110	ERR5 = ***** GX	Q\$CR2 = 176422	Q\$RSET = 000010
BYTE21 = 000025	BYTE73 = 000111	ERR7 = ***** GX	Q\$CLBR = 176424	Q\$SM = 100000
BYTE22 = 000026	BYTE74 = 000112	FIND = ***** GX	Q\$ATTN = 000100	Q\$SP = 000120
BYTE23 = 000027	BYTE75 = 000113	HANG = ***** GX	Q\$BCL = 000001	Q\$SP2 = 000340
BYTE24 = 000030	BYTE76 = 000114	INCVL = ***** GX	Q\$CCCP = 000040	RGD:EN = 000200
BYTE25 = 000031	BYTE77 = 000115	KILL = ***** GX	Q\$CHB = 000400	RGD:VA = 020000
BYTE26 = 000032	BYTE78 = 000116	LBCP = ***** GX	Q\$CHRL = 000200	RTNPT = ***** GX
BYTE27 = 000033	BYTE79 = 000117	LBMRP = ***** GX	Q\$CLR = 000040	SCAN = ***** GX
BYTE28 = 000034	BYTE8 = 000010	LOC:EN = 000100	Q\$CNC = 030000	SEDCS = ***** GX
BYTE29 = 000035	BYTE80 = 000120	LOC:WA = 040000	Q\$CP = 000060	SEQMM = ***** GX
BYTE3 = 000003	BYTE81 = 000121	LOC:WB = 100000	Q\$CPCC = 000010	SEQ:CI = 000010
BYTE30 = 000036	BYTE82 = 000122	LOOP = ***** GX	Q\$CP2 = 000260	S\$CLR = 000000
BYTE31 = 000037	BYTE83 = 000123	MAREN1 = 000001	Q\$CSC = 010000	S\$LA = 000001
BYTE32 = 000040	BYTE84 = 000124	MAREN2 = 004000	Q\$CSEL = 000360	S\$LB = 000005
BYTE33 = 000041	BYTE85 = 000125	MARLOD = 010000	Q\$CSET = 000002	S\$M = 000006
BYTE34 = 000042	BYTE86 = 000126	MAROUT = 000002	Q\$CSP = 020000	S\$N = 000004
BYTE35 = 000043	BYTE87 = 000127	MAR:LO = 002000	Q\$DMA = 000001	S\$SR = 000007
BYTE36 = 000044	BYTE88 = 000130	MAR:OU = 000040	Q\$ENBK = 040000	S\$S1 = 000010
BYTE37 = 000045	BYTE89 = 000131	MBKALL = 001000	Q\$ENOP = 020000	S\$S2 = 000014
BYTE38 = 000046	BYTE9 = 000011	MBKCLK = 000400	Q\$FAL = 004000	TD\$CTR = 176370

MRPR: MACRO: M1110 27-MAR-80 15:11 PAGE 5-6
SYMBOL TABLE

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

TD\$ETW= 176360	T\$ICD= 000040	WORD18= 000044	WORD46= 000134	WORD74= 000224
TD\$INL= 004000	T\$MODE= 004000	WORD19= 000046	WORD47= 000136	WORD75= 000226
TD\$MEM= 000270	T\$OB= 000036	WORD2= 000004	WORD48= 000140	WORD76= 000230
TD\$OAR= 176344	T\$OBE= 004000	WORD20= 000050	WORD49= 000142	WORD77= 000232
TD\$OTR= 176346	T\$OBF= 010000	WORD21= 000052	WORD5= 000012	WORD78= 000234
TD\$QRD= 000274	T\$OBRA= 000034	WORD22= 000054	WORD50= 000144	WORD79= 000236
TD\$SW= 176376	T\$OBWA= 000032	WORD23= 000056	WORD51= 000146	WORD8= 000020
TD\$TAR= 176372	T\$OUTA= 100000	WORD24= 000060	WORD52= 000150	WORD80= 000240
TD\$TAU= 176362	T\$RBD= 000200	WORD25= 000062	WORD53= 000152	WORD81= 000242
TD\$TDR= 176374	T\$RNB= 000040	WORD26= 000064	WORD54= 000154	WORD82= 000244
TD\$TDW= 176364	T\$RSET= 040000	WORD27= 000066	WORD55= 000156	WORD83= 000246
T\$AD= 000020	T\$SC= 000022	WORD28= 000070	WORD56= 000160	WORD84= 000250
T\$BA= 000002	T\$SCLK= 020000	WORD29= 000072	WORD57= 000162	WORD85= 000252
T\$BD= 000010	T\$SEG1= 000000	WORD3= 000006	WORD58= 000164	WORD86= 000254
T\$BSO= 100000	T\$SEG2= 000001	WORD30= 000074	WORD59= 000166	WORD87= 000256
T\$BT= 000020	T\$SEG3= 000002	WORD31= 000076	WORD6= 000014	WORD88= 000260
T\$BTAR= 000030	T\$SO= 000001	WORD32= 000100	WORD60= 000170	WORD89= 000262
T\$BTD= 002000	T\$UBUS= 100000	WORD33= 000102	WORD61= 000172	WORD9= 000022
T\$CD= 000100	T\$1CLK= 000400	WORD34= 000104	WORD62= 000174	WORD90= 000264
T\$CLK= 002000	T\$BBEN= 000020	WORD35= 000106	WORD63= 000176	WORD91= 000266
T\$DISK= 000200	UBD, IN= 000020	WORD36= 000110	WORD64= 000200	WORD92= 000270
T\$DRD= 000004	WORD0= 000000	WORD37= 000112	WORD65= 000202	WORD93= 000272
T\$EMEM= 010000	WORD1= 000002	WORD38= 000114	WORD66= 000204	WORD94= 000274
T\$FSAA= 000000	WORD10= 000024	WORD39= 000116	WORD67= 000206	WORD95= 000276
T\$FSAB= 000004	WORD11= 000026	WORD4= 000010	WORD68= 000210	WORD96= 000300
T\$FSAC= 000014	WORD12= 000030	WORD40= 000120	WORD69= 000212	WORD97= 000302
T\$FSB2= 000010	WORD13= 000032	WORD41= 000122	WORD7= 000016	WORD98= 000304
T\$IB= 000026	WORD14= 000034	WORD42= 000124	WORD70= 000214	WORD99= 000306
T\$IBAR= 000024	WORD15= 000036	WORD43= 000126	WORD71= 000216	WORDVAL= 000310
T\$IBE= 020000	WORD16= 000040	WORD44= 000130	WORD72= 000220	XTREAD= 001000
T\$IBF= 040000	WORD17= 000042	WORD45= 000132	WORD73= 000222	XTURTE= 000400

. ABS. 000000 000
000000 001
MRPR: 001106 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3569 WORDS (14 PAGES)
DYNAMIC MEMORY: 4916 WORDS (18 PAGES)
ELAPSED TIME: 00:00:47
MRPR: MRPR/SP=C20, 1JIM, C20, 1JMRPR

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

57 000060 000564' .WORD RE1MA .
58 000062 115 120 .ASCII /MP/ ;READ MICROPGM ADDR.
59 000064 000612' .WORD RE1MP .
60 000066 123 127 .ASCII /SW/ ;READ UNIBUS STATUS WORD.
61 000070 000644' .WORD RE1SW .
62 000072 104 127 .ASCII /DW/ ;READ UNIBUS DATA WORD.
63 000074 000736' .WORD RE1DW .
64 000004 RE1LN. == <.-RE1TBL>/4
65 ;
66 ;
67 ;
68 ; STORE.
69 ; PERFORM THIRD LEVEL PARSING.
70 ; EG. IN THE COMMAND:
71 ; MR>ST 0 CR.
72 ; PARSE THE '0'
73 ;
74 000076 ST1::
75 000076 CALL FIND ;FIND OPERAND IN COMMAND LINE.
76 000102 103004 BCC 1$ ;OK, CONTINUE.
77 000104 CALL ERR4 ;NOTHING THERE.
78 000110 000167 000354 JMP ST1X ;RETURN TO MRP (PROMPT)
79 000114 1$ CALL PACK ;CONVERT VALUE IN COMMAND LINE TO BINARY
80 000120 103004 BCC 2$ ;INVALID NUMERIC.
81 000122 000167 000336 CALL ERR5
82 000126 000167 000336 JMP ST1X
83 ;
84 ; CONTINUE PARSING THE COMMAND LINE.
85 ; FIND THE REGISTER MNEMONIC.
86 ;
87 000132 2$ CALL FIND ;LOCATE A NON-BLANK.
88 000136 103004 BCC 3$
89 000140 CALL ERR4 ;NOTHING THERE -- ERROR.
90 000144 000167 000320 JMP ST1X
91 ;
92 ; MATCH REGISTER MNEMONIC FROM THE COMMAND LINE AGAINST
93 ; THE TABLE OF VALID MNEMONICS.
94 ;
95 000150 012700 000011 3$ MOV #ST1LN,R0 ;NUMBER OF TABLE ENTRIES.
96 000154 012702 000000' MOV #ST1TBL,R2 ;R2 -> TABLE.
97 000160 CALL SCAN ;MATCH AGAINST COMMAND LINE.
98 000164 103004 BCC 4$ ;OK, CONTINUE.
99 000166 CALL ERR6
100 000172 000167 000272 JMP ST1X
101 ;
102 ; SAVE THE POINTER TO THE ROUTINE ASSOCIATED WITH THE
103 ; REGISTER. R1 -> ROUTINE ADDRESS.
104 ; CALL ROUTINE TO SCAN COMMAND LINE FOR LOOP INDICATOR.
105 ; EG. MR>ST 0 CR L.
106 ; LOOP FLAG WILL BE SET IF INDICATOR IS PRESENT.
107 ; JUMP TO ROUTINE TO LOAD REGISTER.
108 ;
109 000176 010167 000000G 4$ MOV R1,R1RNTPT ;SAVE POINTER TO RTN.
110 000202 CALL LOOPR ;LOOP?.
111 000206 016701 000000G ST1IN: MOV RTNPT,R1 ;POINT TO ROUTINE.
112 000212 000171 000000 JMP @R1 ;EXECUTE ROUTINE.
113 ;

```


Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
114      ;
115      ;
116      ; MRP CONTROL REG.
117      ST1GR::
118      MOV. BINWD,-(SP)      ;MOVE DATA TO LOD BUS REG.
119      CALL. MRPCR.          ;DIRECT CNTL WORD TO MRP.
120      BR. ST1LP.            ;TEST LOOP FLAG.
121      ;
122      ;
123      ; BREAKPOINT REG.
124      ST1BK::
125      MOV. #MBKCLKEN,-(SP)  ;SET MRP CNTL FOR BREAKPOINT.
126      CALL. MRPCR.          ;DIRECT CNTL WORD TO MRP.
127      MOV. BINWD,-(SP)      ;LOAD DATA WORD INTO LOAD BUS REG.
128      CALL. LBMRP.          ;SET BCE CNTL WORD.
129      BR. ST1LP.            ;TEST LOOP FLAG.
130      ;
131      ;
132      ; MEMORY ADDRESS REG.
133      ST1MA::
134      MOV. #<PLR.EN>,-(SP)   ;SET CNTL BITS FOR MRP.
135      CALL. MRPCR.          ;DIRECT CNTL WORD TO MRP.
136      MOV. #<MAR.LOD>,-(SP) ;CNTL BITS TO LOAD MAR.
137      CALL. LBMRP.          ;SEND THEM TO MRP.
138      MOV. #MAREN1,-(SP)    ;CLEAR PLR-RT-ENABLE BITS.
139      CALL. MRPCR.          ;AND SET MAREN1 IN CR.
140      MOV. BINWD,-(SP)      ;MOVE DATA WORD TO LOD BUS REG.
141      CALL. LBMRP.          ;SEND DATA TO MRP (MAR)
142      BR. ST1LP.
143      ;
144      ;
145      ; PIPELINE REG LEFT.
146      ST1PL::
147      MOV. #<PLR.EN+MMLEFT>,-(SP) ;BIT PATTERN FOR PLR LEFT
148      CALL. MRPCR.          ;DIRECT CNTL WORD TO MRP.
149      MOV. BINWD,-(SP)      ;MOVE DATA WORD TO LOD BUS REG.
150      CALL. LBMRP.          ;SEND DATA WORD TO MRP.
151      BR. ST1LP.
152      ;
153      ;
154      ; PIPELINE REG RIGHT.
155      ST1PR::
156      MOV. #<PLR.EN>,-(SP)    ;CNTL WORD FOR PLR RT.
157      CALL. MRPCR.          ;DIRECT CNTL WORD TO MRP.
158      MOV. BINWD,-(SP)      ;MOVE DATA WORD TO LOD BUS REG.
159      CALL. LBMRP.          ;SEND DATA WORD TO MRP.
160      BR. ST1LP.
161      ;
162      ;
163      ; Q-REG.
164      ST1QR::
165      MOV. #<PLR.EN>,-(SP)    ;BIT PATTN FOR PLR RT.
166      CALL. MRPCR.          ;DIRECT CNTL WORD TO MRP.
167      MOV. #RGQ.EN,-(SP)     ;ENABLE Q REG. LOAD DATA WORD TO MRP.
168      CALL. LBMRP.          ;SEND PATTN TO MRP.
169      ;
170      MOV. #<PLR.EN+MMLEFT>,-(SP) ;BIT PATTERN FOR PLR LEFT
```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

171 000404 CALL MRPCR. ;DIRECT CNTL WORD TO MRP.
172 000410 012746 020000 MOV. #RGD.VAL, -(SP) ;BIT PATTN TO GENERATE Q-REG RDY.
173 000414 CALL LBMRP. ;SEND PATTN TO MRP.
174 000420 016746 000000G. MOV. BINWD, -(SP) ;DATA FOR DESTINATION.
175 000424 CALL LBMRP. ;SEND TO MRP.
176 000430 000411 BR STILP.
177 ;
178 ; WORD LOCATION.
179 ;
180 000432 STILW: CALL LOC1W.
181 000432 BR STILP.
182 000436 000406 ;
183 ; SENTENCE LOCATION.
184 ;
185 ;
186 000440 STILS: CALL LOCIS.
187 000440 BR STILP.
188 000444 000403 ;
189 ; PARAGRAPH LOCATION.
190 ;
191 ;
192 000446 STILPG: CALL LOCIP.
193 000446 BR STILP.
194 000452 000400 ;
195 ;
196 ;
197 000454 STILP: BIT #LOOP.BASE. ;LOOP ON COMMAND.
198 000454 032767 000000G-000000G. BEQ STIX ;NO EXIT.
199 000462 001402 JMP STIIN. ;EXECUTE ROUTINE AGAIN.
200 000464 000167 177516 ;
201 ;
202 000470 STIX: CALL KILL ;KILL AST. (IF THERE WAS ONE)
203 000470 JMP MRPXX.
204 000474 000167 000000G.

```

```

206      ;
207      ;
208      ;
209      ;
210      ;
211      ;
212      ;
213      ;
214      ;
215      000500      RE1::
216      000500      CALL      FIND      ;FIND A REGISTER MNEMONIC IN COMMAND LINE.
217      000504      BCC      1$      ;OK, CONTINUE.
218      000506      CALL      ERR4      ;NOTHING THERE.
219      000512      000167      000376      JMP      RE1X      ;RETURN TO MRP (PROMPT)
220      ;
221      ;
222      ;
223      ;
224      000516      012700      000004      1$:
225      000522      012702      000056      MOV      #RE1LN,R0      ;NUMBER OF TABLE ENTRIES.
226      000526      CALL      #RE1TBL,R2      ;R2 -> TABLE.
227      000532      103004      CALL      SCAN      ;MATCH AGAINST COMMAND LINE.
228      000534      BCC      2$      ;OK, CONTINUE.
229      000540      000167      000350      CALL      ERR6
230      000540      JMP      RE1X
231      ;
232      ;
233      ;
234      ;
235      ;
236      ;
237      ;
238      000544      010167      000000G      2$:
239      000550      MOV      R1,RTNPT      ;SAVE POINTER TO RTN
240      000554      016701      000000G      CALL      LOOPR      ;LOOP?
241      000560      000171      000000      MOV      RTNPT,R1      ;POINT TO ROUTINE.
242      000560      JMP      @R1      ;EXECUTE ROUTINE.
243      ;
244      ;
245      ;
246      000564      RE1MA::
247      000564      012746      000040      MOV      #MAR.OUT,-(SP)
248      000570      CALL      MRPCR      ;DIRECT CNTL WORD TO MRP.
249      000574      CALL      MRPLB      ;GET WORD FROM MRP.
250      000600      005046      CLR      -(SP)      ;CLEAR THE CONTROL REG.
251      000602      CALL      MRPCR      ;
252      000606      012601      MOV      (SP)+,R1      ;WORD RETURNED ON STACK.
253      000610      000516      BR      RE1PUT
254      ;
255      ;
256      ;
257      000612      RE1MP::
258      000612      012746      000100      MOV      #MMADR,-(SP)
259      000616      CALL      MRPCR      ;DIRECT CNTL WORD TO MRP.
260      000622      CALL      MRPLB      ;GET WORD FROM MRP.
261      000626      005046      CLR      -(SP)      ;CLEAR THE CONTROL REG.
262      000630      CALL      MRPCR

```

```
263 000634 012601 MOV. (SP)+,R1 ;WORD-RETURNED-ON-STACK
264 000636 042701 177400 BIC. #177400,R1 ;MASK-OFF-UNNECESSARY-BITS
265 000642 000501 BR RE1PUT
266 ;
267 ; UNIBUS-STATUS-REG
268 ;
269 000644 RE1SW: MOV. #<PLR.EN+MMLEFT>,-(SP) ;PL-LEFT NS
270 000644 012746 000202 CALL. MRPCR. ;DIRECT-CNTL-WORD-TO-MRP NS
271 000650 CLR. -(SP) ;DATA-WORD NS
272 000654 005046 CALL. LBMRP. ;SEND-WORD-TO-MRP
273 000656 MOV. #<PLR.EN>,-(SP) ;PL-RIGHT NS
274 000662 012746 000200 CALL. MRPCR. ;DIRECT-CNTL-WORD-TO-MRP
275 000666 MOV. #<UBD.IN+MSYN>,-(SP) ;MASTER-SYNC
276 000672 012746 000060 CALL. LBMRP. ;SEND-WORD-TO-MRP
277 000676 CALL. MRPLB. ;READ-STATUS-WORD-FROM-MRP
278 000702 MOV. #<PLR.EN>,-(SP) ;CLEAR-PLR.EN-BIT NS
279 000706 012746 000200 CALL. MRPCR.
280 000712 CLR. -(SP) ;CLEAR-PLR-RT. NS
281 000716 005046 CALL. LBMRP. ;SEND-WORD-TO-MRP NS
282 000720 CLR. -(SP) ;CLEAR-THE-CR. NS
283 000724 005046 CALL. MRPCR.
284 000726 MOV. (SP)+,R1 ;WORD-RETURNED-ON-STACK
285 000732 012601 BR RE1PUT ;AND-PRINT
286 000734 000444
287 ;
288 ; UNIBUS-DATA-WORD
289 ;
290 000736 RE1DW: MOV. #<PLR.EN+MMLEFT>,-(SP) ;SET-CR-FOR-PL-LEFT NS
291 000736 012746 000202 CALL. MRPCR. ;DIRECT-CNTL-WORD-TO-MRP NS
292 000742 MOV. #01,-(SP) ;SET-DATA-WORD-ADDRESS-BIT NS
293 000746 012746 010000 CALL. LBMRP. ;SEND-WORD-TO-MRP-PLR-LEFT NS
294 000752 MOV. #<PLR.EN>,-(SP) ;SET-CR-FOR-PL-RIGHT NS
295 000756 012746 000200 CALL. MRPCR. ;DIRECT-CNTL-WORD-TO-MRP
296 000762 MOV. #<UBD.IN+MSYN>,-(SP) ;SET-MASTER-SYNC-UBD.IN-IN-PLR-RT. NS
297 000766 012746 000060 CALL. LBMRP. ;SEND-WORD-TO-MRP
298 000772 CALL. MRPLB. ;READ-DATA-WORD-FROM-MRP NS
299 000776 MOV. #<PLR.EN>,-(SP) ;CLEAR-PLR.EN-BIT NS
300 001002 012746 000200 CALL. MRPCR.
301 001006 CLR. -(SP) ;CLEAR-PLR-RT.
302 001012 005046 CALL. LBMRP. ;SEND-WORD-TO-MRP NS
303 001014 MOV. #<MMLEFT+PLR.EN>,-(SP) ;PL-LEFT NS
304 001020 012746 000202 CALL. MRPCR. ;DIRECT-CNTL-WORD-TO-MRP NS
305 001024 CLR. -(SP) ;CLEAR-PLR-LT.
306 001030 005046 CALL. LBMRP. ;CLEAR-PLR-LEFT NS
307 001032 CLR. -(SP) ;CLEAR-MRP-CR
308 001036 005046 CALL. MRPCR.
309 001040 MOV. (SP)+,R1 ;WORD-RETURNED-ON-STACK
310 001044 012601
311 ;
312 ;
313 001046 032767 000000G-000000G RE1PUT: BIT. #ONCE,BASE ;PRINTED-ONCE-?
314 001054 001011 BNE. 1$ ;YES-SKIP
315 001056 052767 000000G-000000G BIS. #ONCE,BASE ;SET-FLAG-FOR-PRINTED-ONCE
316 001064 012705 000000G MOV. #PRINT,R5 ;POINT-TO-PRINT-LINE
317 001070 CALL. UNPK ;CONVERT-VALUE-IN-R1-FOR-PRINTING
318 001074 CALL. CONSOL ;PRINT-ON-CONSOLE
319
```

MRREST. MACRO. M1110 27-MAR-80 15:13 PAGE 6-2.

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

320	001100	032767	000000G-000000G-1\$:	BIT-	#LOOP, BASE-	; REPEAT-
321	001106	001402		BEQ-	RE1X	; NO. EXIT-
322	001110	000167	177440	JMP-	RE1IN-	; EXECUTE ROUTINE AGAIN-
323						
324	001114					
325	001114	042767	000000G-000000G-	BIC-	#ONCE, BASE-	; CLEAR PRINT CONTROL FLAG-
326	001122			CALL-	KILL	; KILL AST-
327	001126	000167	000000G-	JMP-	MRPXX-	

```
329      :  
330      :  
331      : RESET-  
332      :  
333      :  
334 001132 RS1::  
335 001132 005046 CLR - (SP) :CLEAR NOTHING FW  
336 001134 012746 000004 MOV #Q#MSET, -(SP) :SET MRP RESET FW  
337 001140 CALL CSR1 :RESETS MRP FW  
338 001144 012746 000004 MOV #Q#MSET, -(SP) :CLEAR RESET FW  
339 001150 005046 CLR - (SP) :SET NOTHING FW  
340 001152 CALL CSR1 :ELIMINATES MRP RESET FW  
341 001156 000167 000000G JMP MRPXX  
342      :  
343      :  
344      :  
345      : CALL HQR LOADER  
346      :  
347      :  
348 001162 CL1::  
349 001162 CALL CL :CALL ROUTINE IN QMAIN  
350 001166 000167 000000G JMP MRPXX
```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

352. ;
353. ;
354. ; STORE INTO LOCATION WORDS.
355. ;
356. ;
357 001172 005267 176646 LOCIP:: INC. LOCWORD. ; PARAGRAPH
358 001176 005267 176642 LOCIS:: INC. LOCWORD. ; SENTENCE
359 001202. LOCIW:: ; WORD
360 001202 012746 000200 MOV. #<PLR.EN>,-(SP) ; SET BIT PATTN FOR PLR RT
361 001206 CALL. MRPCR. ; DIRECT CNTL WORD TO MRP
362 001212 012746 000100 MOV. #LOC.EN,-(SP) ; SET BIT PATTN FOR LOC ENABLE
363 001216 CALL. LBMRP. ; SEND WORD TO MRP
364. ;
365 001222 012746 000202 MOV. #<PLR.EN+MMLEFT>,-(SP) ; SET BIT PATTN FOR PLR LEFT
366 001226 CALL. MRPCR. ; DIRECT CNTL WORD TO MRP
367 001232 016700 176606 MOV. LOCWORD,R0 ; * DETERMINE WHICH LOCATION
368 001236 006300 ASL. R0 ; * WORD TO SEND TO
369 001240 016046 000046 MOV. LOCTBL(R0),-(SP) ; * THE MRP
370 001244 CALL. LBMRP. ; SEND LOC WORD
371. ;
372 001250 005046 CLR. -(SP) ; DISABLE PIPELIN REGISTER F.W. 8/31
373 001252 CALL. MRPCR.
374 001256 016746 000000 MOV. BINWD,-(SP) ; MOVE USER VALUE TO LOD BUS
375 001262 CALL. LBMRP. ; SEND VALUE TO MRP
376 001266 005067 176552 CLR. LOCWORD. ; RESET
377 001272 RETURN.
378. ;
379 000001 .END.

```

ALUCKE = 040000
ALUOE = 004000
A01 = 010000
BASE = *****
BINWD = *****
BITVAL = 000000
BIT0 = 000001
BIT1 = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2 = 000004
BIT3 = 000010
BIT4 = 000020
BIT5 = 000040
BIT6 = 000100
BIT7 = 000200
BIT8 = 000400
BIT9 = 001000
BYTE0 = 000000
BYTE1 = 000001
BYTE10 = 000012
BYTE11 = 000013
BYTE12 = 000014
BYTE13 = 000015
BYTE14 = 000016
BYTE15 = 000017
BYTE16 = 000020
BYTE17 = 000021
BYTE18 = 000022
BYTE19 = 000023
BYTE2 = 000002
BYTE20 = 000024
BYTE21 = 000025
BYTE22 = 000026
BYTE23 = 000027
BYTE24 = 000030
BYTE25 = 000031
BYTE26 = 000032
BYTE27 = 000033
BYTE28 = 000034
BYTE29 = 000035
BYTE3 = 000003
BYTE30 = 000036
BYTE31 = 000037
BYTE32 = 000040
BYTE33 = 000041
BYTE34 = 000042
BYTE35 = 000043
BYTE36 = 000044
BYTE37 = 000045
BYTE38 = 000046
BYTE39 = 000047
BYTE4 = 000004
BYTE40 = 000050
BYTE41 = 000051
BYTE42 = 000052
BYTE43 = 000053
BYTE44 = 000054
BYTE45 = 000055
BYTE46 = 000056
BYTE47 = 000057
BYTE48 = 000060
BYTE49 = 000061
BYTE5 = 000005
BYTE50 = 000062
BYTE51 = 000063
BYTE52 = 000064
BYTE53 = 000065
BYTE54 = 000066
BYTE55 = 000067
BYTE56 = 000070
BYTE57 = 000071
BYTE58 = 000072
BYTE59 = 000073
BYTE6 = 000006
BYTE60 = 000074
BYTE61 = 000075
BYTE62 = 000076
BYTE63 = 000077
BYTE64 = 000100
BYTE65 = 000101
BYTE66 = 000102
BYTE67 = 000103
BYTE68 = 000104
BYTE69 = 000105
BYTE7 = 000007
BYTE70 = 000106
BYTE71 = 000107
BYTE72 = 000110
BYTE73 = 000111
BYTE74 = 000112
BYTE75 = 000113
BYTE76 = 000114
BYTE77 = 000115
BYTE78 = 000116
BYTE79 = 000117
BYTE8 = 000010
BYTE80 = 000120
BYTE81 = 000121
BYTE82 = 000122
BYTE83 = 000123
BYTE84 = 000124
BYTE85 = 000125
BYTE86 = 000126
BYTE87 = 000127
BYTE88 = 000130
BYTE89 = 000131
BYTE9 = 000011
BYTE90 = 000132
BYTE91 = 000133
BYTE92 = 000134
BYTE93 = 000135
BYTE94 = 000136
BYTE95 = 000137
BYTE96 = 000140
BYTE97 = 000141
BYTE98 = 000142
BYTE99 = 000143
BYTVAL = 000144
CBKALL = 001000
CBKCLK = 000400
CL = ***** GX
CL1 = 001162RG
CNOBRE = 100000
CONSOL = ***** GX
CPCCEN = 010000
CPREAD = 040000
CPWRITE = 020000
CSADRD = 000004
CSEDCI = 100000
CSOE = 000040
CSR1 = ***** GX
CSWRITE = 000100
DBR.RD = 000001
DB\$CPP = 001457
DB\$SPT = 000026
DB\$TPC = 000023
DISPGS = 100000
DMAAUR = 000005
DMARRD = 000003
DMARWR = 000004
ENBR = 010000
ERR4 = ***** GX
EPRS = ***** GX
ERR6 = ***** GX
FIND = ***** GX
KILL = ***** GX
LBMRP = ***** GX
LOCTBL = 000046R
LOCURD = 000044R
LOC.EN = 000100
LOC.WA = 040000
LOC.WB = 100000
LOCIP = 001172RG
LOCIS = 001176RG
LOCIW = 001202RG
LOOP = ***** GX
LOOPR = ***** GX
MAREN1 = 000001
MAREN2 = 004000
MARLOD = 010000
MAROUT = 000002
MAR.LO = 002000
MAR.OU = 000040
MBKALL = 001000
MEKCLK = 000400
MMADRD = 000100
MMLEFT = 000002
MMOE = 000004
MMWRTE = 000010
MNOBRE = 100000
MREN1 = 000001
MREN2 = 020000
MRPCR = ***** GX
MRPLB = ***** GX
MRPXX = ***** GX
MSYN = 000040
N = 000144
ONCE = ***** GX
002.PACK = ***** GX
PLB = 000010
PLC = 000020
PLD = 000030
PLRWR = 000200
PLR.EN = 000200 GX
PRINT = ***** GX
QR\$CR1 = 176420
QR\$CR2 = 176422
QR\$LBR = 176424
Q\$ATTN = 000100
Q\$BCL = 000001
Q\$CCCP = 000040
Q\$CHB = 000400
Q\$CHRL = 000200
Q\$CLR = 000040
Q\$CNC = 030000
Q\$CP = 000060
Q\$CPCC = 000010
Q\$CP2 = 000260
Q\$CSC = 010000
Q\$CSEL = 000360
Q\$CSET = 000002
Q\$CSP = 020000
Q\$DMA = 000001
Q\$ENBK = 040000
002.Q\$ENOP = 020000
002.Q\$FAL = 004000
Q\$FC = 000045
Q\$FO = 000044
Q\$FP = 000046
002.Q\$HBF = 000002
002.Q\$ICP = 000006
002.Q\$IHB = 000003
Q\$IHRL = 000002
Q\$IMRP = 000007
Q\$LBD = 001000
Q\$LBPD = 001001
Q\$LBP = 000001
Q\$LCD = 000003
Q\$LMD = 000004
Q\$LDPP = 002000
Q\$LHP = 010000
Q\$INC = 140000
Q\$MR = 000052
Q\$MRP = 000040
Q\$MRP2 = 000240
Q\$MSC = 040000
Q\$MSET = 000004
Q\$MSP = 100000
Q\$NCLK = 176000
Q\$PP = 000100
Q\$PPSW = 000320
Q\$PP2 = 000300
Q\$QHLT = 000013
Q\$QL = 000043
Q\$QLA = 000053
Q\$QLB = 000054
Q\$QLR = 000001
Q\$QW = 000042
Q\$RDCD = 000005
Q\$RDMD = 000006
Q\$REBK = 001000
Q\$RNC = 006000
Q\$RSC = 004000
Q\$RSET = 000010
Q\$SM = 100000
Q\$SP = 000120
Q\$SP2 = 000340
REI = 000500RG 002
REIDW = 000736RG 002
REIIN = 000554R 002
REILN = 000004 G
REIMA = 000564RG 002
REIMP = 000612RG 002
REIPUT = 001046R 002
REISW = 000644RG 002
REITBL = 000056RG 002
REIX = 001114R 002
RGQ.EN = 000200
RGQ.VA = 020000
RSI = 001132RG 002
RTNPT = ***** GX
SCAN = ***** GX
SEQ.CI = 000010
STI = 000076RG 002
STIBK = 000230RG 002
STICR = 000216RG 002
STIIN = 000206R 002
STILN = 000011 G
STILP = 000454R 002
STILPG = 000446RG 002
STILS = 000440RG 002
STILW = 000432RG 002
STIMA = 000252RG 002
STIPL = 000314RG 002
STIPR = 000366RG 002
STIQR = 000360RG 002
STITBL = 000000RG 002
STIX = 000470R 002
S\$CLR = 000000
S\$LA = 000001

MRREST: MACRO M1110 27-MAR-80 15:13 PAGE 8-2
SYMBOL TABLE

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

S#QB = 000005	T#FSB2 = 000010	WORD14 = 000034	WORD44 = 000130	WORD73 = 000222
S#QR = 000006	T#IB = 000026	WORD15 = 000036	WORD45 = 000132	WORD74 = 000224
S#QX = 000004	T#IBAR = 000024	WORD16 = 000040	WORD46 = 000134	WORD75 = 000226
S#SR = 000007	T#IBE = 020000	WORD17 = 000042	WORD47 = 000136	WORD76 = 000230
S#S1 = 000010	T#IBF = 040000	WORD18 = 000044	WORD48 = 000140	WORD77 = 000232
S#S2 = 000014	T#ICD = 000040	WORD19 = 000046	WORD49 = 000142	WORD78 = 000234
TD#CTR = 176370	T#MODE = 004000	WORD2 = 000004	WORD5 = 000012	WORD79 = 000236
TD#CTW = 176360	T#OB = 000036	WORD20 = 000050	WORD50 = 000144	WORD8 = 000020
TD#INL = 004000	T#OBE = 004000	WORD21 = 000052	WORD51 = 000146	WORD80 = 000240
TD#MEM = 000270	T#OBF = 010000	WORD22 = 000054	WORD52 = 000150	WORD81 = 000242
TD#OAR = 176344	T#OBRA = 000034	WORD23 = 000056	WORD53 = 000152	WORD82 = 000244
TD#OTR = 176346	T#OBWA = 000032	WORD24 = 000060	WORD54 = 000154	WORD83 = 000246
TD#QRD = 000274	T#OUTA = 100000	WORD25 = 000062	WORD55 = 000156	WORD84 = 000250
TD#SW = 176376	T#RBD0 = 000200	WORD26 = 000064	WORD56 = 000160	WORD85 = 000252
TD#TAR = 176372	T#RNB = 000040	WORD27 = 000066	WORD57 = 000162	WORD86 = 000254
TD#TAW = 176362	T#RSET = 040000	WORD28 = 000070	WORD58 = 000164	WORD87 = 000256
TD#TDR = 176374	T#SC = 000022	WORD29 = 000072	WORD59 = 000166	WORD88 = 000260
TD#TDW = 176364	T#SCLK = 020000	WORD3 = 000006	WORD6 = 000014	WORD89 = 000262
T#AD = 000020	T#SEG1 = 000000	WORD30 = 000074	WORD60 = 000170	WORD9 = 000022
T#BA = 000002	T#SEG2 = 000001	WORD31 = 000076	WORD61 = 000172	WORD90 = 000264
T#BD = 000010	T#SEG3 = 000002	WORD32 = 000100	WORD62 = 000174	WORD91 = 000266
T#BS0 = 100000	T#SO = 000001	WORD33 = 000102	WORD63 = 000176	WORD92 = 000270
T#BT = 000020	T#UBUS = 100000	WORD34 = 000104	WORD64 = 000200	WORD93 = 000272
T#BTAR = 000030	T#1CLK = 000400	WORD35 = 000106	WORD65 = 000202	WORD94 = 000274
T#BTD = 002000	T#BBEN = 000020	WORD36 = 000110	WORD66 = 000204	WORD95 = 000276
T#CD = 000100	UBD, IN = 000020	WORD37 = 000112	WORD67 = 000206	WORD96 = 000300
T#CLK = 002000	UNPK = ***** GX	WORD38 = 000114	WORD68 = 000210	WORD97 = 000302
T#DISK = 000200	WORD0 = 000000	WORD39 = 000116	WORD69 = 000212	WORD98 = 000304
T#DRD = 000004	WORD1 = 000002	WORD4 = 000010	WORD7 = 000016	WORD99 = 000306
T#EMEM = 010000	WORD10 = 000024	WORD40 = 000120	WORD70 = 000214	WORDVAL = 000310
T#FSAA = 000000	WORD11 = 000026	WORD41 = 000122	WORD71 = 000216	XTREAD = 001000
T#FSAB = 000004	WORD12 = 000030	WORD42 = 000124	WORD72 = 000220	XTWRITE = 000400
T#FSAC = 000014	WORD13 = 000032	WORD43 = 000126		

. ABS. 000000 000
000000 001
MRREST: 001274 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3285 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:47
MRREST, MRREST--SP=[20,1]IM,[20,1]MRREST

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

1      .TITLE MRBUG
2 000000 .PSECT MRBUG
3
4
5
6      HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS
7      MATCH REPORT PROCESSOR TEST ROUTINES
8
9      DEBUGGING COMMANDS
10     COMMANDS:
11     AT      SET BREAKPOINT
12     OF      REMOVE BREAKPOINT
13     SS      SINGLE STEP
14     GO      PROCEED FROM BREAKPOINT
15
16     ONCE A COMMAND HAS BEEN EXECUTED (OR AN ERROR ENCOUNTERED)
17     THIS MODULE RETURNS CONTROL TO THE MODULE MRP AT LOCATION
18     'MRPXX'
19
20
21     *****
22
23     DISCLAIMER:
24     IN ORDER FOR THE BREAKPOINT ROUTINES TO WORK PROPERLY
25     THERE MUST BE A 'PIPELINE REGISTER ONLY' CLOCK. THIS
26     WOULD ALLOW THE CHANGING OF PLR CONTENTS WITHOUT CAUSING
27     AN INSTRUCTION TO BE EXECUTED. AS IT STANDS NOW, THE PIPE-
28     LINE REGISTER CLOCK IS TIED TO THE SEQUENCER CLOCK. WHEN
29     A BREAKPOINT HAS BEEN REACHED, ANY OPERATION INVOLVING
30     A PLR CLOCK (EG. PRINTING MICROBGM MEMORY) WILL EXECUTE
31     THE INSTRUCTION IN THE PLR. THE 2900 SEQUENCER STACK IS
32     ALSO SOMETIMES CHANGED.
33
34     *****
35
36     .MCALL CLEF$S, WTLO$S, RDAF$S, WTSE$S
37
38
39     SET BREAKPOINT
40     EG. MR>AT 22
41
42 AT1::
43 000000 CALL FIND ;LOOK FOR BKPT ADDRESS IN COMMAND LINE
44 000004 103004 BCC 1$ ;OK, CONVERT BKPT ADDRESS
45 000006 CALL ERR4 ;NOTHING THERE
46 000012 000167 000000G JMP MRPXX
47 000016 1$ CALL PACK ;CONVERT BKPT ADDRESS
48 000022 103004 BCC 2$ ;OK, CONTINUE
49 000024 CALL ERR5 ;ERROR ON CONVERSION
50 000030 000167 000000G JMP MRPXX
51 000034 026767 000000G-000000G-2$ CMP MMHIGH, BINWD ;IS ADDRESS IN RANGE
52 000042 003004 BGT 3$ ;YES, CONTINUE
53 000044 CALL ERR10 ;ADDR OUT OF RANGE
54 000050 000167 000000G JMP MRPXX
55
56 000054 012746 000400 3$ MOV #MBKCLKEN, -(SP) ;SET MRP CNTL FOR BREAKPOINT
57 000060 CALL MRPCR ;DIRECT CNTL WORD TO MRP

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
58 000064 016746 000000G . . . . . MOV. BINWD, -(SP) ;LOAD BKPT ADDR INTO LOAD BUS REG.
59 000070 CALL. LBMRP ;LOAD MRP BKPT REG.
60 000074 052767 000000G 000000G BIS. #BREAK, BASE ;SET FLAG FOR BKPT SET.
61 000102 000167 000000G JMP. MRPXX
62. ;
63. ;
64. ; REMOVE BREAKPOINT (DISABLE)
65. ; MR>OF.
66. ;
67. ;
68 000106 OF1::
69 000106 005046 CLR. -(SP) ;SEND 0 TO MRP CONTROL REG.
70 000110 CALL. MRPCR ;EXECUTE TRANSFER.
71 000114 005067 176422 CLR. QR$CR2 ;CLEAR CSR 2.
72 000120 042767 000000G 000000G BIC. #BREAK, BASE ;CLEAR BREAKPOINT SET FLAG.
73 000126 000167 000000G JMP. MRPXX
74. ;
75. ;
76. ; SINGLE STEP.
77. ; EG. MR>SS.
78. ; MR>SS 22
79. ;
80. ;
81 000132 SS1::
82 000132 CALL. FIND ;LOOK FOR ADDR IN COMMAND LINE
83 000136 103426 BCS. 3$ ;NONE THERE, USE CURRENT MAR.
84 000140 CALL. PACK ;CONVERT START ADDRESS.
85 000144 103004 BCC. 1$ ;OK, CONTINUE.
86 000146 CALL. ERR5 ;ERROR ON CONVERSION
87 000152 000167 000000G JMP. MRPXX
88. ;
89 000156 026767 000000G 000000G 1$: CMP. MMHIGH, BINWD ;IS ADDRESS IN RANGE
90 000164 003004 BGT. 2$ ;YES, CONTINUE.
91 000166 CALL. ERR10 ;OUT OF RANGE HIGH.
92 000172 000167 000000G JMP. MRPXX
93. ;
94. ; SEQUENCE UP TO SINGLE STEP ADDRESS (IF NECESSARY)
95. ; SET SEARCH MODE.
96. ; SINGLE CLOCK ALL TO EXECUTE ONE INSTRUCTION.
97. ; SET LOAD MODE.
98. ;
99 000176 016746 000000G 2$: MOV. BINWD, -(SP) ;PROVIDE START ADDRESS FOR SEQUENCER.
100 000202 CALL. SEQMM ;SEQUENCE ONLY UP TO START ADDRESS.
101 000206 005046 CLR. -(SP) ;REINHIBIT BRANCH CONTROL REG.
102 000210 CALL. MRPCR
103 000214 3$:
104 000214 012746 001777 MOV. #1777, -(SP) ;SET CP MICROCODE TO INNOCUOUS LOC (X'3FF')
105 000220 CALL. SEQCS ;
106 000224 005046 CLR. -(SP) ;REINHIBIT BR CONTROL REG.
107 000226 CALL. CPCR
108. ;
109 000232 012767 100000 176422 MOV. #0$SM, QR$CR2 ;SET SEARCH MODE.
110 000240 012746 146000 MOV. *(<Q$INC+Q$RNC>), -(SP) ;CLEAR MRP AND PPS 40-CLOCKS.
111 000244 052716 001001 BIS. *(<Q$LBD+Q$LBP>), (SP) ;CLR DRIVE & PULSE.
112 000250 012746 100000 MOV. #Q$MSP, -(SP) ;SET SINGLE CLOCK ALL.
113 000254 CALL. CSR1 ;MOVE TO CSR1
114 000260 005046 CLR. -(SP) ;CLEAR NOTHING.
```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

115 000262 012746 146000      MOV.    *(<Q$MNC+Q$RNC>)-(SP)    ;SET MRP AND PPS NO-CLOCKS.
116 000266                      CALL.    CSR1                ;AND REWRITE CSR1
117                               ;
118 000272 005067 176422      CLR.    QR$CR2.          ;SET LOAD MODE.
119 000276                      CALL.    MRPMP.             ;PRINT MICROPGM ADDRESS.
120 000302 000167 000000G.    JMP.    MRPXX.
121                               ;
122                               ;
123                               ;
124                               ;
125                               ;
126                               ;
127                               ;
128 000306                      ;
129 000306                      G01::
130 000312 103431      CALL.    FIND                ;LOOK FOR START ADDRESS.
131 000314                      BCS.    MGO.             ;NONE, GO FROM PRESENT ADDRESS
132 000320 103004      CALL.    PACK                ;CONVERT START ADDRESS.
133 000322                      BCC.    1$              ;OK, CONTINUE.
134 000326 000167 000000G.    CALL.    ERR5           ;ERROR ON CONVERSION
135                               ;
136 000332 026767 000000G-000000G-1$: CMP.    MMHIGH,BINWD.    ;IS START ADDRESS IN RANGE.
137 000340 003004                      BGT.    2$              ;YES, CONTINUE.
138 000342                      CALL.    ERR10          ;OUT OF RANGE HIGH.
139 000346 000167 000000G.    JMP.    MRPXX.
140                               ;
141 000352 016746 000000G.    2$: MOV.    BINWD, -(SP)      ;PROVIDE START ADDRESS FOR SEQUENCER.
142 000356                      CALL.    SEQMM.          ;SEQUENCE ONLY UP TO START ADDRESS.
143                               ;
144 000362 005046                      CLR.    -(SP)        ;REINHIBIT BRANCH CONTROL REG.
145 000364                      CALL.    MRPCR.
146 000370 005046                      CLR.    -(SP)
147 000372                      CALL.    LBMRP.          ;SINGLE CLOCK TO REINHIBIT BRANCH REGISTER.
148                               ;
149                               ;
150                               ;
151                               ;
152 000376                      ;
153 000376 012746 001777      MGO:
154 000402                      MOV.    #1777, -(SP)      ;SET CP MICROCODE TO INNOCUOUS LOC (X'3FF')
155 000406 005046                      CALL.    SEQCS.
156 000410                      CLR.    -(SP)
157                               ;
158 000414 032767 000000G-000000G.    CALL.    CPCR
159 000422 001047                      ;
160 000424 012767 100000 176422.    BIT.    #BREAK,BASE.    ;PROCEED TO BREAKPOINT.
161 000432 012746 146000      MOV.    MBK.             ;YES, SET UP FOR BKPT.
162 000436 052716 001001      MOV.    #Q$SM,QR$CR2.      ;SET SEARCH MODE.
163 000442 005046                      MOV.    *(<Q$MNC+Q$RNC>)-(SP) ;CLEAR MRP AND PPS NO-CLOCKS.
164 000444                      BIS.    *(<Q$LBD+Q$LBP>),(SP) ;CLEAR DRIVE AND PULSE.
165                               ;
166                               ;
167                               ;
168                               ;
169 000450                      CLR.    -(SP)              ;SET NOTHING.
170                               ;
171 000462                      CALL.    CSR1              ;NEW CSR1
172                               ;
173                               ;
174                               ;
175                               ;
176                               ;
177                               ;
178                               ;
179                               ;
180                               ;
181                               ;
182                               ;
183                               ;
184                               ;
185                               ;
186                               ;
187                               ;
188                               ;
189                               ;
190                               ;
191                               ;
192                               ;
193                               ;
194                               ;
195                               ;
196                               ;
197                               ;
198                               ;
199                               ;
200                               ;
201                               ;
202                               ;
203                               ;
204                               ;
205                               ;
206                               ;
207                               ;
208                               ;
209                               ;
210                               ;
211                               ;
212                               ;
213                               ;
214                               ;
215                               ;
216                               ;
217                               ;
218                               ;
219                               ;
220                               ;
221                               ;
222                               ;
223                               ;
224                               ;
225                               ;
226                               ;
227                               ;
228                               ;
229                               ;
230                               ;
231                               ;
232                               ;
233                               ;
234                               ;
235                               ;
236                               ;
237                               ;
238                               ;
239                               ;
240                               ;
241                               ;
242                               ;
243                               ;
244                               ;
245                               ;
246                               ;
247                               ;
248                               ;
249                               ;
250                               ;
251                               ;
252                               ;
253                               ;
254                               ;
255                               ;
256                               ;
257                               ;
258                               ;
259                               ;
260                               ;
261                               ;
262                               ;
263                               ;
264                               ;
265                               ;
266                               ;
267                               ;
268                               ;
269                               ;
270                               ;
271                               ;
272                               ;
273                               ;
274                               ;
275                               ;
276                               ;
277                               ;
278                               ;
279                               ;
280                               ;
281                               ;
282                               ;
283                               ;
284                               ;
285                               ;
286                               ;
287                               ;
288                               ;
289                               ;
290                               ;
291                               ;
292                               ;
293                               ;
294                               ;
295                               ;
296                               ;
297                               ;
298                               ;
299                               ;
300                               ;
301                               ;
302                               ;
303                               ;
304                               ;
305                               ;
306                               ;
307                               ;
308                               ;
309                               ;
310                               ;
311                               ;
312                               ;
313                               ;
314                               ;
315                               ;
316                               ;
317                               ;
318                               ;
319                               ;
320                               ;
321                               ;
322                               ;
323                               ;
324                               ;
325                               ;
326                               ;
327                               ;
328                               ;
329                               ;
330                               ;
331                               ;
332                               ;
333                               ;
334                               ;
335                               ;
336                               ;
337                               ;
338                               ;
339                               ;
340                               ;
341                               ;
342                               ;
343                               ;
344                               ;
345                               ;
346                               ;
347                               ;
348                               ;
349                               ;
350                               ;
351                               ;
352                               ;
353                               ;
354                               ;
355                               ;
356                               ;
357                               ;
358                               ;
359                               ;
360                               ;
361                               ;
362                               ;
363                               ;
364                               ;
365                               ;
366                               ;
367                               ;
368                               ;
369                               ;
370                               ;
371                               ;
372                               ;
373                               ;
374                               ;
375                               ;
376                               ;
377                               ;
378                               ;
379                               ;
380                               ;
381                               ;
382                               ;
383                               ;
384                               ;
385                               ;
386                               ;
387                               ;
388                               ;
389                               ;
390                               ;
391                               ;
392                               ;
393                               ;
394                               ;
395                               ;
396                               ;
397                               ;
398                               ;
399                               ;
400                               ;
401                               ;
402                               ;
403                               ;
404                               ;
405                               ;
406                               ;
407                               ;
408                               ;
409                               ;
410                               ;
411                               ;
412                               ;
413                               ;
414                               ;
415                               ;
416                               ;
417                               ;
418                               ;
419                               ;
420                               ;
421                               ;
422                               ;
423                               ;
424                               ;
425                               ;
426                               ;
427                               ;
428                               ;
429                               ;
430                               ;
431                               ;
432                               ;
433                               ;
434                               ;
435                               ;
436                               ;
437                               ;
438                               ;
439                               ;
440                               ;
441                               ;
442                               ;
443                               ;
444                               ;
445                               ;
446                               ;
447                               ;
448                               ;
449                               ;
450                               ;
451                               ;
452                               ;
453                               ;
454                               ;
455                               ;
456                               ;
457                               ;
458                               ;
459                               ;
460                               ;
461                               ;
462                               ;
463                               ;
464                               ;
465                               ;
466                               ;
467                               ;
468                               ;
469                               ;
470                               ;
471                               ;
472                               ;
473                               ;
474                               ;
475                               ;
476                               ;
477                               ;
478                               ;
479                               ;
480                               ;
481                               ;
482                               ;
483                               ;
484                               ;
485                               ;
486                               ;
487                               ;
488                               ;
489                               ;
490                               ;
491                               ;
492                               ;
493                               ;
494                               ;
495                               ;
496                               ;
497                               ;
498                               ;
499                               ;
500                               ;
501                               ;
502                               ;
503                               ;
504                               ;
505                               ;
506                               ;
507                               ;
508                               ;
509                               ;
510                               ;
511                               ;
512                               ;
513                               ;
514                               ;
515                               ;
516                               ;
517                               ;
518                               ;
519                               ;
520                               ;
521                               ;
522                               ;
523                               ;
524                               ;
525                               ;
526                               ;
527                               ;
528                               ;
529                               ;
530                               ;
531                               ;
532                               ;
533                               ;
534                               ;
535                               ;
536                               ;
537                               ;
538                               ;
539                               ;
540                               ;
541                               ;
542                               ;
543                               ;
544                               ;
545                               ;
546                               ;
547                               ;
548                               ;
549                               ;
550                               ;
551                               ;
552                               ;
553                               ;
554                               ;
555                               ;
556                               ;
557                               ;
558                               ;
559                               ;
560                               ;
561                               ;
562                               ;
563                               ;
564                               ;
565                               ;
566                               ;
567                               ;
568                               ;
569                               ;
570                               ;
571                               ;
572                               ;
573                               ;
574                               ;
575                               ;
576                               ;
577                               ;
578                               ;
579                               ;
580                               ;
581                               ;
582                               ;
583                               ;
584                               ;
585                               ;
586                               ;
587                               ;
588                               ;
589                               ;
590                               ;
591                               ;
592                               ;
593                               ;
594                               ;
595                               ;
596                               ;
597                               ;
598                               ;
599                               ;
600                               ;
601                               ;
602                               ;
603                               ;
604                               ;
605                               ;
606                               ;
607                               ;
608                               ;
609                               ;
610                               ;
611                               ;
612                               ;
613                               ;
614                               ;
615                               ;
616                               ;
617                               ;
618                               ;
619                               ;
620                               ;
621                               ;
622                               ;
623                               ;
624                               ;
625                               ;
626                               ;
627                               ;
628                               ;
629                               ;
630                               ;
631                               ;
632                               ;
633                               ;
634                               ;
635                               ;
636                               ;
637                               ;
638                               ;
639                               ;
640                               ;
641                               ;
642                               ;
643                               ;
644                               ;
645                               ;
646                               ;
647                               ;
648                               ;
649                               ;
650                               ;
651                               ;
652                               ;
653                               ;
654                               ;
655                               ;
656                               ;
657                               ;
658                               ;
659                               ;
660                               ;
661                               ;
662                               ;
663                               ;
664                               ;
665                               ;
666                               ;
667                               ;
668                               ;
669                               ;
670                               ;
671                               ;
672                               ;
673                               ;
674                               ;
675                               ;
676                               ;
677                               ;
678                               ;
679                               ;
680                               ;
681                               ;
682                               ;
683                               ;
684                               ;
685                               ;
686                               ;
687                               ;
688                               ;
689                               ;
690                               ;
691                               ;
692                               ;
693                               ;
694                               ;
695                               ;
696                               ;
697                               ;
698                               ;
699                               ;
700                               ;
701                               ;
702                               ;
703                               ;
704                               ;
705                               ;
706                               ;
707                               ;
708                               ;
709                               ;
710                               ;
711                               ;
712                               ;
713                               ;
714                               ;
715                               ;
716                               ;
717                               ;
718                               ;
719                               ;
720                               ;
721                               ;
722                               ;
723                               ;
724                               ;
725                               ;
726                               ;
727                               ;
728                               ;
729                               ;
730                               ;
731                               ;
732                               ;
733                               ;
734                               ;
735                               ;
736                               ;
737                               ;
738                               ;
739                               ;
740                               ;
741                               ;
742                               ;
743                               ;
744                               ;
745                               ;
746                               ;
747                               ;
748                               ;
749                               ;
750                               ;
751                               ;
752                               ;
753                               ;
754                               ;
755                               ;
756                               ;
757                               ;
758                               ;
759                               ;
760                               ;
761                               ;
762                               ;
763                               ;
764                               ;
765                               ;
766                               ;
767                               ;
768                               ;
769                               ;
770                               ;
771                               ;
772                               ;
773                               ;
774                               ;
775                               ;
776                               ;
777                               ;
778                               ;
779                               ;
780                               ;
781                               ;
782                               ;
783                               ;
784                               ;
785                               ;
786                               ;
787                               ;
788                               ;
789                               ;
790                               ;
791                               ;
792                               ;
793                               ;
794                               ;
795                               ;
796                               ;
797                               ;
798                               ;
799                               ;
800                               ;
801                               ;
802                               ;
803                               ;
804                               ;
805                               ;
806                               ;
807                               ;
808                               ;
809                               ;
810                               ;
811                               ;
812                               ;
813                               ;
814                               ;
815                               ;
816                               ;
817                               ;
818                               ;
819                               ;
820                               ;
821                               ;
822                               ;
823                               ;
824                               ;
825                               ;
826                               ;
827                               ;
828                               ;
829                               ;
830                               ;
831                               ;
832                               ;
833                               ;
834                               ;
835                               ;
836                               ;
837                               ;
838                               ;
839                               ;
840                               ;
841                               ;
842                               ;
843                               ;
844                               ;
845                               ;
846                               ;
847                               ;
848                               ;
849                               ;
850                               ;
851                               ;
852                               ;
853                               ;
854                               ;
855                               ;
856                               ;
857                               ;
858                               ;
859                               ;
860                               ;
861                               ;
862                               ;
863                               ;
864                               ;
865                               ;
866                               ;
867                               ;
868                               ;
869                               ;
870                               ;
871                               ;
872                               ;
873                               ;
874                               ;
875                               ;
876                               ;
877                               ;
878                               ;
879                               ;
880                               ;
881                               ;
882                               ;
883                               ;
884                               ;
885                               ;
886                               ;
887                               ;
888                               ;
889                               ;
890                               ;
891                               ;
892                               ;
893                               ;
894                               ;
895                               ;
896                               ;
897                               ;
898                               ;
899                               ;
900                               ;
901                               ;
902                               ;
903                               ;
904                               ;
905                               ;
906                               ;
907                               ;
908                               ;
909                               ;
910                               ;
911                               ;
912                               ;
913                               ;
914                               ;
915                               ;
916                               ;
917                               ;
918                               ;
919                               ;
920                               ;
921                               ;
922                               ;
923                               ;
924                               ;
925                               ;
926                               ;
927                               ;
928                               ;
929                               ;
930                               ;
931                               ;
932                               ;
933                               ;
934                               ;
935                               ;
936                               ;
937                               ;
938                               ;
939                               ;
940                               ;
941                               ;
942                               ;
943                               ;
944                               ;
945                               ;
946                               ;
947                               ;
948                               ;
949                               ;
950                               ;
951                               ;
952                               ;
953                               ;
954                               ;
955                               ;
956                               ;
957                               ;
958                               ;
959                               ;
960                               ;
961                               ;
962                               ;
963                               ;
964                               ;
965                               ;
966                               ;
967                               ;
968                               ;
969                               ;
970                               ;
971                               ;
972                               ;
973                               ;
974                               ;
975                               ;
976                               ;
977                               ;
978                               ;
979                               ;
980                               ;
981                               ;
982                               ;
983                               ;
984                               ;
985                               ;
986                               ;
987                               ;
988                               ;
989                               ;
990                               ;
991                               ;
992                               ;
993                               ;
994                               ;
995                               ;
996                               ;
997                               ;
998                               ;
999                               ;
1000                              ;

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
172 000466      ;      WTSE$S. #EFN.4      ;WAIT FOR QIO COMPLETION (CHAR INTERRUPT)
173      ;
174 000500      ;      CALL. KILL      ;KILL QIO
175 000504 005046      ;      CLR. -(SP)      ;CLEAR NOTHING
176 000506 012746 146000      ;      MOV. #<Q$MNC+Q$RNC>,-(SP)      ;SET MRP AND PPS NO-CLOCKS
177 000512      ;      CALL. CSR1      ;REWRITE CSR1
178 000516 005046      ;      CLR. -(SP)      ;SEND 0 TO MRP CONTROL REG
179 000520      ;      CALL. MRPCR      ;DO IT
180 000524 012767 000000 176422      ;      MOV. #0,QR$CR2      ;SET LOAD MODE
181 000532      ;      CALL. MRPMP      ;PRINT MICROPGM ADDRESS
182 000536 000167 000000G      ;      JMP. MRPXX
183      ;
184      ;
185      ;      EXPECT TO HIT A BREAKPOINT.
186      ;
187 000542      ;      MBK:
188 000542      ;      CLEF$S. #EFN.3      ;CLEAR BKPT EVENT FLAG
189      ;
190 000554      ;      CLEF$S. #EFN.4      ;CLEAR QIO EVENT FLAG
191      ;
192      ;      SET BKPT ALLOW BIT IN MRP CONTROL REG
193      ;      SET SEARCH MODE
194      ;      SET FREE RUN CLOCK
195      ;      WAIT FOR BKPT INTERRUPT THROUGH VECTOR ADDRESS 274 (SEE
196      ;      RTN 'BPTISR') OR CHARACTER INTERRUPT FROM TERMINAL
197      ;
198 000566 012746 001000      ;      MOV. #MBKALLOW,-(SP)      ;SET MRP CNTL WORD
199 000572      ;      CALL. MRPCR      ;DIRECT CNTL WORD TO MRP
200 000576 012767 140000 176422      ;      MOV. #<Q$SM+Q$ENBK>,QR$CR2      ;SET SEARCH MODE
201 000604 012746 146000      ;      MOV. #<Q$MNC+Q$RNC>,-(SP)      ;CLEAR MRP AND PPS NO-CLOCKS
202 000610 052716 001001      ;      BIS. #<Q$LBD+Q$LBP>,(SP)      ;CLEAR DRIVE AND PULSE
203 000614 005046      ;      CLR. -(SP)      ;SET NOTHING
204 000616      ;      CALL. CSR1      ;NEW CSR1
205 000622      ;      CALL. HANG2      ;ISSUE QIO TO TERMINAL
206      ;
207      ;      WAIT FOR INTERRUPT FROM TERMINAL OR BREAKPOINT INTERRUPT
208      ;      FROM MRP
209      ;
210 000626      ;      WTLO$S. 0,#000014      ;WAIT FOR BKPT OR CHAR INTERRUPT
211      ;
212 000642      ;      RDAF$S. #EFBUF      ;READ FOR DEBUGGING
213      ;
214      ;      INTERRUPT RECEIVED
215      ;      RESET NO-CLOCKS IN CSR #1
216      ;      SET LOAD MODE
217      ;      PRINT MICROPGM ADDRESS
218      ;
219 000654 005046      ;      CLR. -(SP)      ;CLEAR NOTHING
220 000656 012746 146000      ;      MOV. #<Q$MNC+Q$RNC>,-(SP)      ;SET MRP AND PPS NO-CLOCKS
221 000662      ;      CALL. CSR1      ;REWRITE CSR1
222 000666 012746 000000      ;      MOV. #0,-(SP)      ;SEND 0 TO MRP CONTROL REG
223 000672      ;      CALL. MRPCR      ;DO IT
224      ;
225 000676 012767 000000 176422      ;      MOV. #0,QR$CR2      ;SET LOAD MODE
226 000704 012767 001000 176422      ;      MOV. #Q$REBK,QR$CR2      ;RE-ARM INTERRUPTS
227 000712      ;      CALL. MRPMP      ;PRINT MICROPGM ADDRESS
228 000716      ;      CALL. KILL      ;KILL QIO
```

MRBUG. MACRO-M1110 27-MAR-80 15:09 PAGE 5-4

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

229 000722. 000167 000000G.

JMP. MRPXX.

```

231      ;
232      ;
233      ; PRINT MICROPROGRAM ADDRESS.
234      ;
235 000726      MRPMP:
236 000726      012746 000100      MOV.    #MMADDR, -(SP)      ; READ MM ADDRESS SELECT BIT.
237 000732      CALL.   MRPCR.      ; WRITE MRP CONTROL REGISTER.
238 000736      CALL.   MRPLB.      ; NOW GET WORD FROM MRP.
239 000742      012601      MOV.    (SP)+, R1      ; LOAD R1 FOR CONVERSION.
240 000744      042701 177400      BIC.    #177400, R1      ; CLEAR UPPER 6 BITS.
241 000750      012705 000000G.    MOV.    #PRINT, R5      ; R5 -> PRINT LINE.
242 000754      CALL.   UNPK.      ; CONVERT MPA.
243 000760      CALL.   CONSOL.    ; PRINT ON CONSOLE.
244 000764      RETURN.
245      ;
246      000001      .END.

```

ALUCKE=.040000	BYTE39=.000047	BYTE90=.000132	MMADDR=.000100	Q\$LHP=.010000
ALUOE=.004000	BYTE4=.000004	BYTE91=.000133	MMHIGH=.000000 GX.	Q\$MNC=.140000
AT1=.000000RG. 002	BYTE40=.000050	BYTE92=.000134	MMLEFT=.000002	Q\$MR=.000052
A01=.010000	BYTE41=.000051	BYTE93=.000135	MMOE=.000004	Q\$MRP=.000040
BASE=.000000 GX.	BYTE42=.000052	BYTE94=.000136	MMWRTE=.000010	Q\$MRP2=.000240
BINWD=.000000 GX.	BYTE43=.000053	BYTE95=.000137	MNOBRE=.100000	Q\$MSC=.040000
BITVAL=.000000	BYTE44=.000054	BYTE96=.000140	MREN1=.000001	Q\$MSET=.000004
BIT0=.000001	BYTE45=.000055	BYTE97=.000141	MREN2=.020000	Q\$MSP=.100000
BIT1=.000002	BYTE46=.000056	BYTE98=.000142	MRPCR=.000000 GX.	Q\$NCLK=.176000
BIT10=.002000	BYTE47=.000057	BYTE99=.000143	MRPLB=.000000 GX.	Q\$PP=.000100
BIT11=.004000	BYTE48=.000058	BYTVAL=.000144	MRPMP=.000726R.	002.Q\$PPSW=.000320
BIT12=.010000	BYTE49=.000059	CBKALL=.001000	MRPXX=.000000 GX.	Q\$PP2=.000300
BIT13=.020000	BYTE5=.000005	CBKCLK=.000400	MSYN=.000040	Q\$QHLT=.000013
BIT14=.040000	BYTE50=.000062	CNOBRE=.100000	N=.000144	Q\$QL=.000043
BIT15=.100000	BYTE51=.000063	CONSOL=.000000 GX.	OF1=.000106RG.	002.Q\$QLA=.000053
BIT2=.000004	BYTE52=.000064	CPCCEN=.010000	PACK=.000000 GX.	Q\$QLB=.000054
BIT3=.000010	BYTE53=.000065	CPCR=.000000 GX.	PLB=.000010	Q\$QLR=.000001
BIT4=.000020	BYTE54=.000066	CFREAD=.040000	PLC=.000020	Q\$QW=.000042
BIT5=.000040	BYTE55=.000067	CPWRTE=.020000	PLD=.000030	Q\$RDCD=.000005
BIT6=.000100	BYTE56=.000070	CSADRD=.000004	PLRWR=.000200	Q\$RDMD=.000006
BIT7=.000200	BYTE57=.000071	CSEOC1=.100000	PLR.EN=.000200	Q\$REBK=.001000
BIT8=.000400	BYTE58=.000072	CSDOE=.000040	PRINT=.000000 GX.	Q\$RNC=.006000
BIT9=.001000	BYTE59=.000073	CSR1=.000000 GX.	QR\$CR1=.176420	Q\$RSC=.004000
BREAK=.000000 GX.	BYTE6=.000006	CSWRTE=.000100	QR\$CR2=.176422	Q\$RSET=.000010
BYTE0=.000000	BYTE60=.000074	DBR.RD=.000001	Q\$CLBR=.176424	Q\$SM=.100000
BYTE1=.000001	BYTE61=.000075	DB\$CPP=.001457	Q\$ATTN=.000100	Q\$SP=.000120
BYTE10=.000012	BYTE62=.000076	DB\$SPT=.000026	Q\$BCL=.000001	Q\$SP2=.000340
BYTE11=.000013	BYTE63=.000077	DB\$TPC=.000023	Q\$CCCP=.000040	RGQ.EN=.000200
BYTE12=.000014	BYTE64=.000100	DISPGS=.100000	Q\$CHB=.000400	RGQ.VA=.020000
BYTE13=.000015	BYTE65=.000101	DMAWR=.000005	Q\$CHRL=.000200	SEQCS=.000000 GX.
BYTE14=.000016	BYTE66=.000102	DMARRD=.000003	Q\$CLR=.000040	SEQMM=.000000 GX.
BYTE15=.000017	BYTE67=.000103	DMARWR=.000004	Q\$CNC=.030000	SEQ.CI=.000010
BYTE16=.000020	BYTE68=.000104	EFBUF=.000000 GX.	Q\$CP=.000060	SS1=.000132RG. 002.
BYTE17=.000021	BYTE69=.000105	EFN.3=.000000 GX.	Q\$CPCC=.000010	S\$CLR=.000000
BYTE18=.000022	BYTE7=.000007	EFN.4=.000000 GX.	Q\$CP2=.000260	S\$LA=.000001
BYTE19=.000023	BYTE70=.000106	ENBR=.010000	Q\$CSC=.010000	S\$OB=.000005
BYTE2=.000002	BYTE71=.000107	ERR10=.000000 GX.	Q\$CSEL=.000360	S\$OR=.000006
BYTE20=.000024	BYTE72=.000110	ERR4=.000000 GX.	Q\$CSET=.000002	S\$OX=.000004
BYTE21=.000025	BYTE73=.000111	ERR5=.000000 GX.	Q\$CSP=.020000	S\$SR=.000007
BYTE22=.000026	BYTE74=.000112	FIND=.000000 GX.	Q\$DMA=.000001	S\$S1=.000010
BYTE23=.000027	BYTE75=.000113	GO1=.000306RG. 002	Q\$ENBK=.040000	S\$S2=.000014
BYTE24=.000030	BYTE76=.000114	HANG2=.000000 GX.	Q\$ENOP=.020000	TD\$CTR=.176370
BYTE25=.000031	BYTE77=.000115	KILL=.000000 GX.	Q\$FAL=.004000	TD\$CTW=.176360
BYTE26=.000032	BYTE78=.000116	LBMRP=.000000 GX.	Q\$FC=.000045	TD\$INL=.004000
BYTE27=.000033	BYTE79=.000117	LOC.EN=.000100	Q\$FD=.000044	TD\$MEM=.000270
BYTE28=.000034	BYTE8=.000010	LOC.WA=.040000	Q\$FP=.000046	TD\$OAR=.176344
BYTE29=.000035	BYTE80=.000120	LOC.WB=.100000	Q\$HBF=.000002	TD\$OTR=.176346
BYTE3=.000003	BYTE81=.000121	MAREN1=.000001	Q\$ICP=.000006	TD\$QRD=.000274
BYTE30=.000036	BYTE82=.000122	MAREN2=.004000	Q\$IH=.000003	TD\$SW=.176376
BYTE31=.000037	BYTE83=.000123	MARLOD=.010000	Q\$IHRL=.000002	TD\$TAR=.176372
BYTE32=.000040	BYTE84=.000124	MAROUT=.000002	Q\$IMRP=.000007	TD\$TAU=.176362
BYTE33=.000041	BYTE85=.000125	MAR.LO=.002000	Q\$LBD=.001000	TD\$TDR=.176374
BYTE34=.000042	BYTE86=.000126	MAR.OU=.000040	Q\$LBDP=.001001	TD\$TDW=.176364
BYTE35=.000043	BYTE87=.000127	MBK=.000542R.	002.Q\$LBP=.000001	T\$AD=.000000
BYTE36=.000044	BYTE88=.000130	MBKALL=.001000	Q\$LDCL=.000005	T\$BA=.000002
BYTE37=.000045	BYTE89=.000131	MBKCLK=.000400	Q\$LDID=.000004	T\$BD=.000010
BYTE38=.000046	BYTE9=.000011	MGO=.000376R.	002.Q\$LDPP=.002000	T\$BSO=.100000

MRBUG: M1110 27-MAR-80 15:09 PAGE 6-2
SYMBOL: TA

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

T\$BT= .000020	T\$SEG1= .000000	WORD26= .000064	WORD51= .000146	WORD77= .000232
T\$BTAR= .000030	T\$SEG2= .000001	WORD27= .000066	WORD52= .000150	WORD78= .000234
T\$BTD= .002000	T\$SEG3= .000002	WORD28= .000070	WORD53= .000152	WORD79= .000236
T\$CD= .000100	T\$SO= .000001	WORD29= .000072	WORD54= .000154	WORD8= .000020
T\$CLK= .002000	T\$UBUS= .100000	WORD3= .000006	WORD55= .000156	WORD80= .000240
T\$DISK= .000200	T\$1CLK= .000400	WORD30= .000074	WORD56= .000160	WORD81= .000242
T\$DRD= .000004	T\$BBEN= .000020	WORD31= .000076	WORD57= .000162	WORD82= .000244
T\$EMEM= .010000	UBD: IN= .000020	WORD32= .000100	WORD58= .000164	WORD83= .000246
T\$FSAA= .000000	UNPK= .***** GX	WORD33= .000102	WORD59= .000166	WORD84= .000250
T\$FSAB= .000004	WORD0= .000000	WORD34= .000104	WORD6= .000014	WORD85= .000252
T\$FSAC= .000014	WORD1= .000002	WORD35= .000106	WORD60= .000170	WORD86= .000254
T\$FSB2= .000016	WORD10= .000024	WORD36= .000110	WORD61= .000172	WORD87= .000256
T\$IB= .000026	WORD11= .000026	WORD37= .000112	WORD62= .000174	WORD88= .000260
T\$IBAR= .000024	WORD12= .000030	WORD38= .000114	WORD63= .000176	WORD89= .000262
T\$IBE= .000000	WORD13= .000032	WORD39= .000116	WORD64= .000200	WORD9= .000022
T\$IBF= .000000	WORD14= .000034	WORD4= .000010	WORD65= .000202	WORD90= .000264
T\$ICD= .000040	WORD15= .000036	WORD40= .000120	WORD66= .000204	WORD91= .000266
T\$MODE= .004000	WORD16= .000040	WORD41= .000122	WORD67= .000206	WORD92= .000270
T\$OB= .000036	WORD17= .000042	WORD42= .000124	WORD68= .000210	WORD93= .000272
T\$OBE= .004000	WORD18= .000044	WORD43= .000126	WORD69= .000212	WORD94= .000274
T\$OBF= .010000	WORD19= .000046	WORD44= .000130	WORD7= .000016	WORD95= .000276
T\$OBRA= .000034	WORD2= .000004	WORD45= .000132	WORD70= .000214	WORD96= .000300
T\$OBWA= .000032	WORD20= .000050	WORD46= .000134	WORD71= .000216	WORD97= .000302
T\$OUTA= .100000	WORD21= .000052	WORD47= .000136	WORD72= .000220	WORD98= .000304
T\$RBD0= .000200	WORD22= .000054	WORD48= .000140	WORD73= .000222	WORD99= .000306
T\$RNB= .000040	WORD23= .000056	WORD49= .000142	WORD74= .000224	WRDVAL= .000310
T\$RSET= .000000	WORD24= .000060	WORD5= .000012	WORD75= .000226	XTREAD= .001000
T\$SC= .000022	WORD25= .000062	WORD50= .000144	WORD76= .000230	XTWRTE= .000400
T\$SCLK= .020000				

. ABS. 000000 000
000000 001
MRBUG: 000766 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3584 WORDS (14 PAGES)
DYNAMIC MEMORY: 4916 WORDS (18 PAGES)
ELAPSED TIME: 00:00:48
MRBUG, MRBUG/SP=[20, 1]IM, [20, 1]MRBUG

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
1      .TITLE GR...
2      .PSECT CP
3      .LIST MEB
4      ;
5      ;
6      ;
7      ;
8      ;
9      ;
10     ;
11     ;
12     ;
13     ;
14     ;
15     ;
16     ;
17     ;
18     ;
19     ;
20     ;
21     ;
22     ;
23     ;
24     ;
25     ;
26     ;
27     ;
28     ;
29     ;
30     ;
31     ;
32     ;
33     ;
34     ;
35     ;
36     ;
37     ;
```

HARDWARE QUERY RESOLVER. "MANUAL" DEBUGGING AIDS. CONTROL PROCESSOR TEST ROUTINES.

PARSE THE COMMAND LINE AND TRANSFER CONTROL TO ONE OF THE CP SUB-MODULES OR BACK TO QMAIN. CONTROL COULD HAVE BEEN PASSED TO CP IN ONE OF THREE WAYS:

1. FROM QMAIN IF CP WAS SELECTED AS THE FIRST PROCESSOR. UPON ENTRY TO THE PROGRAM.
CP>LD CD 0
2. FROM QMAIN IF CP WAS SELECTED FROM ONE OF QMAIN'S OTHER SUB-MODULES.
MR>CP LD CD 0
3. ON RETURN FROM ONE OF CP'S SUB-MODULES.
CP>LD CD 0

CP PARSES AT THE SECOND LEVEL OF CONTROL (SEE NOTES AT QMAIN ON LEVELS OF CONTROL). IN THE EXAMPLES ABOVE, CP WOULD PARSE 'LD' AND TRANSFER CONTROL TO THE CP SUB-MODULE CPLD. IF CP SHOULD ENCOUNTER A STRING WHICH IS NOT A VALID COMMAND MNEMONIC, CP CONSIDERS THE STRING TO BE A PROCESSOR MNEMONIC AND RETURNS CONTROL TO QMAIN. EG: CP>MR LD MM 0
THE STRING 'MR' IS NOT A VALID CP (SECOND LEVEL) COMMAND. CP RETURNS CONTROL TO QMAIN WHICH IN TURN WILL TRANSFER CONTROL TO ITS SUB-MODULE MRP.

CP SUB-MODULES:

- CP LD LOAD MEMORIES.
- CP PR PRINT MEMORY CONTENTS.
- CP RES REMAINING COMMANDS (EXCEPT DEBUGGING COMMANDS)
- CP BUG1 DEBUGGING COMMANDS.
- CP BUG2 DEBUGGING COMMANDS.

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

39      :
40      :
41      :
42      :
43      :
44      :
45      :
46      :
47      :
48      :
49      :
50      :
51      :
52      :
53      :
54      :
55      :
56      :
57      :
58      :
59      :
60      :
61      :
62      :
63      :
64      :
65      :
66      :
67      :
68      :
69      :
70      :
71      :
72      :
73      :
74      :
75      :
76      :

```

000000	101	124	CPTBL:	.ASCII /AT/	:SET BREAKPOINT
000002	000000G			.WORD AT2	
000004	117	106		.ASCII /OF/	:DISABLE BREAKPOINT
000006	000000G			.WORD OF2	
000010	114	122		.ASCII /LR/	:LOAD ALL REGISTERS
000012	000000G			.WORD LR2	
000014	122	107		.ASCII /RG/	:PRINT REGISTERS
000016	000000G			.WORD RG2	
000020	120	103		.ASCII /PC/	:SINGLE STEP A NUMBER OF TIMES (PROCEED)
000022	000000G			.WORD PC2	
000024	123	123		.ASCII /SS/	:SINGLE STEP
000026	000000G			.WORD SS2	
000030	107	117		.ASCII /GO/	:PROCEED FROM BREAKPOINT
000032	000000G			.WORD GO2	
000034	106	122		.ASCII /FR/	:RUN BOTH MRP AND CP
000036	000000G			.WORD FR2	
000040	123	124		.ASCII /ST/	:STORE INTO A REG
000042	000000G			.WORD ST2	
000044	122	105		.ASCII /RE/	:READ FROM A REG
000046	000000G			.WORD RE2	
000050	114	104		.ASCII /LD/	:LOAD MEMORY
000052	000000G			.WORD LD2	
000054	120	122		.ASCII /PR/	:PRINT FROM MEMORY
000056	000000G			.WORD PR2	
000060	120	102		.ASCII /PB/	:PRINT BUFFERS
000062	000000G			.WORD PB2	
000064	122	123		.ASCII /RS/	
000066	000000G			.WORD RS2	
000070	103	114		.ASCII /CL/	:CALL HQR LOADER
000072	000000G			.WORD CL2	
000017			CPNUM:	=	<.-CPTBL>/4

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

78
79
80
81 000074 000000
82
83
84
85 000076
86 000076 000200
87 000100 000130
88 000102 000000
89 000104 000000
90 000106 000000
91 000110 000530
92 000112 000000
93 000114 000000
94 000116 000000
95 000120
96
97
98
99 000130
100 000130
101 000530
102 000530 000001
103 000532 021042
104 000534 031464
105 000536 000401
106
107 000540 000004
108 000542 052525
109 000544 063146
110 000546 000402
111 000550 000002
112
113 000552 000007
114 000554 104210
115 000556 114631
116 000560 000403
117 000562 001402
118
119 000564 000012
120 000566 135673
121 000570 146314
122 000572 000401
123 000574 140000
124 000576
125
126

```

```

:
: INDEX WORD FOR READING AND WRITING CP REGISTERS
:
INDEX:: .WORD 0
:
: BCL
:
BCL::
: .WORD 128. :BUFFERS ARE 128 WORDS
: .WORD HLB :HIT LIST BUFFER
: .WORD 0
: .WORD 0
: .WORD 0
: .WORD HRL :HRL BUFFER
: .WORD 0
: .WORD 0
: .WORD 0
: .BLKW 4 :END OF BUFFER ADDRESSES
:
: HIT LIST AND HRL BUFFERS
:
HLB::
: .BLKW 128.
HRL::
: .WORD 000001 :=X'0001'
: .WORD 021042 :=X'2222'
: .WORD 031464 :=X'3334'
: .WORD 000401 :=X'0101'
:
: .WORD 000004 :=X'0004'
: .WORD 052525 :=X'5555'
: .WORD 063146 :=X'6666'
: .WORD 000402 :=X'0102'
: .WORD 000002 :=X'0002'
:
: .WORD 000007 :=X'0007'
: .WORD 104210 :=X'8888'
: .WORD 114631 :=X'9999'
: .WORD 000403 :=X'0103'
: .WORD 001402 :=X'0302'
:
: .WORD 000012 :=X'000A'
: .WORD 135673 :=X'BBBB'
: .WORD 146314 :=X'CCCC'
: .WORD 000401 :=X'0101'
: .WORD 140000 :=X'C000'
: .BLKW 128.-<.-HRL/2>
: .LIST BEX
: .NLIST CND

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

128.      ;
129.      ;
130.      ;      CP ROUTINES.
131.      ;
132.      ;
133.001130      CP::
134.001130      004767      000000G      JSR      PC,FIND      ;LOCATE THE COMMAND IN THE COMMAND LINE.
135.001134      103003      BCC      1$      ;OK, CONTINUE.
136.001136      004767      000000G      JSR      PC,ERR3
137.001142      000422      BR      CPXX
138.001144      022700      000002      1$: CMP      #2,R0      ;COMMANDS ARE 2 CHARS.
139.001150      001403      BEQ      2$
140.001152      004767      000000G      JSR      PC,ERR8
141.001156      000414      BR      CPXX      ;TRY AGAIN.
142.      ;
143.001160      012700      000017      2$: MOV      #CPNUM,R0      ;R0 = NUMBER OF COMMANDS.
144.001164      012702      000000G      MOV      #CPTBL,R2      ;R2 -> TABLE OF COMMAND MNEMONICS.
145.001170      004767      000000G      JSR      PC,SCAN      ;FIND MATCH IN TABLE.
146.001174      103003      BCC      3$      ;OK, CONTINUE.
147.001176      004767      000000G      JSR      PC,ERR12      ;COMMAND NOT IN TABLE.
148.001202      000402      BR      CPXX      ;TRY AGAIN.
149.      ;
150.      ;
151.      ;      JUMP TO THE ROUTINE THAT GOVERNS THE COMMAND.
152.001204      000171      000000      3$: JMP      @ (R1)
153.      ;
154.      ;
155.      ;      LOCAL CP LOOP, INCLUDE PROMPT FOR CP COMMAND.
156.      ;
157.      ;
158.001210      CPXX::
159.001210      012767      050103      000000G      MOV      #"CP,GCMBLK+G,DPRM+2 ;MOVE CP NAME TO GCM BLOCK.
160.001216      004767      000000G      JSR      PC,GCONLY      ;PROMPT.
161.001222      004767      000000G      JSR      PC,FIND      ;LOCATE THE COMMAND IN THE COMMAND LINE.
162.001226      103003      BCC      1$      ;OK, CONTINUE.
163.001230      004767      000000G      JSR      PC,ERR3
164.001234      000765      BR      CPXX
165.001236      022700      000002      1$: CMP      #2,R0      ;COMMANDS ARE 2 CHARS.
166.001242      001403      BEQ      2$
167.001244      004767      000000G      JSR      PC,ERR8
168.001250      000757      BR      CPXX      ;TRY AGAIN.
169.      ;
170.001252      012700      000017      2$: MOV      #CPNUM,R0      ;R0 = NUMBER OF COMMANDS.
171.001256      012702      000000G      MOV      #CPTBL,R2      ;R2 -> TABLE OF COMMAND MNEMONICS.
172.001262      004767      000000G      JSR      PC,SCAN      ;FIND MATCH IN TABLE.
173.001266      103005      BCC      3$      ;OK, CONTINUE.
174.001270      042767      000000G-000000G      BIC      #BREAK,BASE      ;CLEAR BREAKPOINT FLAG.
175.001276      000167      000000G      JMP      COMXX      ;RETURN TO QMAIN, LOOK FOR PROCESSOR MNEMONIC.
176.      ;
177.      ;      JUMP TO THE ROUTINE THAT GOVERNS THE COMMAND.
178.      ;
179.001302      000171      000000      3$: JMP      @ (R1)

```

```

181      ;
182      ;
183      ; *****
184      ;
185      ;     TEMPORARILY DELETED
186      ;
187      ; *****
188      ;
189      ;
190      ;     LOAD A CP INTERNAL REGISTER
191      ;
192      ;     UPON ENTRY:
193      ;     R0 = REGISTER NUMBER
194      ;     THE FIELD 'BINWD' CONTAINS THE BIT PATTERN TO BE LOADED
195      ;
196      ;
197 001306 LDREG:
198 001306 000207      RTS      PC

```

```
200      ;
201      ;
202      ; *****
203      ;
204      ;     TEMPORARILY DELETED
205      ;
206      ; *****
207      ;
208      ;
209      ;     PRINT CP REGISTERS
210      ;
211      ;     UPON ENTRY:
212      ;     R0 = REGISTER TO BE READ
213      ;
214      ;     UPON EXIT:
215      ;     R1 = REGISTER VALUE
216      ;
217      ;
218 001310 REREG::
219 001310 000207     RTS     PC
220
221     000001     .END
```

ALUCKE = 040000
ALUOE = 004000
AT2 = ***** GX
A01 = 010000
BASE = ***** GX
BCL = 000076RG 002
BITVAL = 000000
BIT0 = 000001
BIT1 = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2 = 000004
BIT3 = 000010
BIT4 = 000020
BIT5 = 000040
BIT6 = 000100
BIT7 = 000200
BIT8 = 000400
BIT9 = 001000
BREAK = ***** GX
BYTE0 = 000000
BYTE1 = 000001
BYTE10 = 000012
BYTE11 = 000013
BYTE12 = 000014
BYTE13 = 000015
BYTE14 = 000016
BYTE15 = 000017
BYTE16 = 000020
BYTE17 = 000021
BYTE18 = 000022
BYTE19 = 000023
BYTE2 = 000002
BYTE20 = 000024
BYTE21 = 000025
BYTE22 = 000026
BYTE23 = 000027
BYTE24 = 000030
BYTE25 = 000031
BYTE26 = 000032
BYTE27 = 000033
BYTE28 = 000034
BYTE29 = 000035
BYTE3 = 000003
BYTE30 = 000036
BYTE31 = 000037
BYTE32 = 000040
BYTE33 = 000041
BYTE34 = 000042
BYTE35 = 000043
BYTE36 = 000044
BYTE37 = 000045
BYTE38 = 000046
BYTE39 = 000047
BYTE4 = 000004
BYTE40 = 000050
BYTE41 = 000051
BYTE42 = 000052
BYTE43 = 000053
BYTE44 = 000054
BYTE45 = 000055
BYTE46 = 000056
BYTE47 = 000057
BYTE48 = 000060
BYTE49 = 000061
BYTE5 = 000005
BYTE50 = 000062
BYTE51 = 000063
BYTE52 = 000064
BYTE53 = 000065
BYTE54 = 000066
BYTE55 = 000067
BYTE56 = 000070
BYTE57 = 000071
BYTE58 = 000072
BYTE59 = 000073
BYTE6 = 000006
BYTE60 = 000074
BYTE61 = 000075
BYTE62 = 000076
BYTE63 = 000077
BYTE64 = 000100
BYTE65 = 000101
BYTE66 = 000102
BYTE67 = 000103
BYTE68 = 000104
BYTE69 = 000105
BYTE7 = 000007
BYTE70 = 000106
BYTE71 = 000107
BYTE72 = 000110
BYTE73 = 000111
BYTE74 = 000112
BYTE75 = 000113
BYTE76 = 000114
BYTE77 = 000115
BYTE78 = 000116
BYTE79 = 000117
BYTE8 = 000010
BYTE80 = 000120
BYTE81 = 000121
BYTE82 = 000122
BYTE83 = 000123
BYTE84 = 000124
BYTE85 = 000125
BYTE86 = 000126
BYTE87 = 000127
BYTE88 = 000130
BYTE89 = 000131
BYTE9 = 000011
BYTE90 = 000132
BYTE91 = 000133
BYTE92 = 000134
BYTE93 = 000135
BYTE94 = 000136
BYTE95 = 000137
BYTE96 = 000140
BYTE97 = 000141
BYTE98 = 000142
BYTE99 = 000143
BYTVAL = 000144
CBKALL = 001000
CBKCLK = 000400
CL2 = ***** GX
CNOBRE = 100000
COMXX = ***** GX
CP = 001130RG 002
CPCCEN = 010000
CPNUM = 000017
CPREAD = 040000
CPTBL = 000000R
CPWRTE = 020000
CPXX = 001210RG 002
CSADRD = 000004
CSEQCI = 100000
CSOE = 000040
CSWRTE = 000100
DBR, RD = 000001
DB\$CPP = 001457
DB\$SPT = 000026
DB\$TPC = 000023
DISPGS = 100000
DMAAUR = 000005
DMARRD = 000003
DMARWR = 000004
ENBR = 010000
ERR12 = ***** GX
ERR3 = ***** GX
ERR8 = ***** GX
FIND = ***** GX
FR2 = ***** GX
GCMBLK = ***** GX
GCONLY = ***** GX
GO2 = ***** GX
G, DPRM = ***** GX
HLB = 000130RG 002
HRL = 000530RG 002
INDEX = 000074RG 002
LDREG = 001306RG 002
LD2 = ***** GX
LOC, EN = 000100
LOC, WA = 040000
LOC, WB = 100000
LR2 = ***** GX
MAREN1 = 000001
MAREN2 = 000000
MARLOD = 010000
MAROUT = 000002
MAR, LO = 002000
MAR, OU = 000040
MBKALL = 001000
MBKCLK = 000400
MMADDR = 000100
MMLEFT = 000002
MMOE = 000004
MMWRTE = 000010
MNOBRE = 100000
MREN1 = 000001
MREN2 = 020000
MSYN = 000040
N = 000144
OF2 = ***** GX
PB2 = ***** GX
PC2 = ***** GX
PLB = 000010
PLC = 000020
PLD = 000030
PLRWR = 000200
PLR, EN = 000200
PR2 = ***** GX
QR\$CR1 = 176420
QR\$CR2 = 176422
QR\$LBR = 176424
Q\$ATTN = 000100
Q\$BCL = 000001
Q\$CCCP = 000040
Q\$CHB = 000400
Q\$CHRL = 000200
Q\$CLR = 000040
Q\$CNC = 030000
Q\$CP = 000060
Q\$CPCC = 000010
Q\$CP2 = 000260
Q\$CSC = 010000
Q\$CSEL = 000360
Q\$CSET = 000002
Q\$CSP = 020000
Q\$DMA = 000001
Q\$ENBK = 040000
Q\$ENOP = 020000
Q\$FAL = 004000
Q\$FC = 000045
Q\$FO = 000044
Q\$FP = 000045
Q\$HBF = 000002
Q\$ICP = 000006
Q\$IHB = 000003
Q\$IHRL = 000002
Q\$IMRP = 000007
Q\$LBD = 001000
Q\$LBDP = 001001
Q\$LBP = 000001
Q\$LCD = 000003
Q\$LMD = 000004
Q\$LDPP = 002000
Q\$LHP = 010000
Q\$MNC = 140000
Q\$MR = 000052
Q\$MRP = 000040
Q\$MRP2 = 000240
Q\$MSC = 040000
Q\$MSET = 000004
Q\$MSP = 100000
Q\$NCLK = 176000
Q\$PP = 000100
Q\$PPSW = 000320
Q\$PP2 = 000300
Q\$QHLT = 000013
Q\$QL = 000043
Q\$QLA = 000053
Q\$QLB = 000054
Q\$QLR = 000001
Q\$QW = 000042
Q\$RDCD = 000005
Q\$RDM = 000006
Q\$REBK = 001000
Q\$RNC = 006000
Q\$RSC = 004000
Q\$RSET = 000010
Q\$SM = 100000
Q\$SP = 000120
Q\$SP2 = 000340
REREG = 001310RG 002
RE2 = ***** GX
RGQ, EN = 000200
RGQ, VA = 020000
RG2 = ***** GX
RS2 = ***** GX
SCAN = ***** GX
SEQ, CI = 000010
SS2 = ***** GX
ST2 = ***** GX
S\$CLR = 000000
S\$LA = 000001
S\$OB = 000005
S\$OR = 000006
S\$OX = 000004
S\$SR = 000007
S\$S1 = 000010
S\$S2 = 000014
TD\$CTR = 176370
TD\$CTW = 176360
TD\$INL = 004000
TD\$MEM = 000270
TD\$OAR = 176344
TD\$QTR = 176346
TD\$RDR = 000274
TD\$SW = 176376
TD\$TAR = 176372
TD\$TAW = 176362
TD\$TDR = 176374

CP.....MACRO:M1110 27-MAR-80 14:40 PAGE 10-3
SYMBOL TABLE

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

TD\$TDW= 176364	T\$RBD0= 000200	WORD23= 000056	WORD5 = 000012	WORD76= 000230
T\$AD= 000020	T\$RNB= 000040	WORD24= 000060	WORD50= 000144	WORD77= 000232
T\$BA= 000002	T\$RSET= 040000	WORD25= 000062	WORD51= 000146	WORD78= 000234
T\$BD= 000010	T\$SC= 000022	WORD26= 000064	WORD52= 000150	WORD79= 000236
T\$BS0= 100000	T\$SCLK= 020000	WORD27= 000066	WORD53= 000152	WORD8 = 000020
T\$BT= 000020	T\$SEG1= 000000	WORD28= 000070	WORD54= 000154	WORD80= 000240
T\$BTAR= 000030	T\$SEG2= 000001	WORD29= 000072	WORD55= 000156	WORD81= 000242
T\$BTD= 002000	T\$SEG3= 000002	WORD3 = 000006	WORD56= 000160	WORD82= 000244
T\$CD= 000100	T\$SO= 000001	WORD30= 000074	WORD57= 000162	WORD83= 000246
T\$CLK= 002000	T\$UBUS= 100000	WORD31= 000076	WORD58= 000164	WORD84= 000250
T\$DISK= 000200	T\$1CLK= 000400	WORD32= 000100	WORD59= 000166	WORD85= 000252
T\$DRD= 000004	T\$BBEN= 000020	WORD33= 000102	WORD6 = 000014	WORD86= 000254
T\$EMEM= 010000	UBD, IN= 000020	WORD34= 000104	WORD60= 000170	WORD87= 000256
T\$FSA= 000000	WORD0 = 000000	WORD35= 000106	WORD61= 000172	WORD88= 000260
T\$FSAB= 000004	WORD1 = 000002	WORD36= 000110	WORD62= 000174	WORD89= 000262
T\$FSAC= 000014	WORD10= 000024	WORD37= 000112	WORD63= 000176	WORD9 = 000022
T\$FSB2= 000010	WORD11= 000026	WORD38= 000114	WORD64= 000200	WORD90= 000264
T\$IB= 000026	WORD12= 000030	WORD39= 000116	WORD65= 000202	WORD91= 000266
T\$IBAR= 000024	WORD13= 000032	WORD4 = 000010	WORD66= 000204	WORD92= 000270
T\$IBE= 020000	WORD14= 000034	WORD40= 000120	WORD67= 000206	WORD93= 000272
T\$IBF= 040000	WORD15= 000036	WORD41= 000122	WORD68= 000210	WORD94= 000274
T\$ICD= 000040	WORD16= 000040	WORD42= 000124	WORD69= 000212	WORD95= 000276
T\$MODE= 004000	WORD17= 000042	WORD43= 000126	WORD7 = 000016	WORD96= 000300
T\$OB= 000036	WORD18= 000044	WORD44= 000130	WORD70= 000214	WORD97= 000302
T\$OBE= 004000	WORD19= 000046	WORD45= 000132	WORD71= 000216	WORD98= 000304
T\$OBF= 010000	WORD2 = 000004	WORD46= 000134	WORD72= 000220	WORD99= 000306
T\$OBRA= 000034	WORD20= 000050	WORD47= 000136	WORD73= 000222	WORDVAL= 000310
T\$OBWA= 000032	WORD21= 000052	WORD48= 000140	WORD74= 000224	XTREAD= 001000
T\$OUTA= 100000	WORD22= 000054	WORD49= 000142	WORD75= 000226	XTWRTE= 000400

. ABS. 000000 000
000000 001
CP. 001312 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3201 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:42
CP*GP*-SP=C20.1JIM,C20.1JCP

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

1          .TITLE CPBUG1
2 000000    .PSECT CPBUG1
3          ;
4          ;
5          ;   HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS.
6          ;   ***** PROTOTYPE VERSION *****
7          ;   CONTROL PROCESSOR TEST ROUTINES.
8          ;
9          ;   COMMANDS:
10         ;   AT   SET BREAKPOINT
11         ;   OF   REMOVE BREAKPOINT
12         ;   LR   LOAD MICROPROCESSOR REGISTERS
13         ;   RG   PRINT MICROPROCESSOR REGISTERS
14         ;   PC   SINGLE STEP A NUMBER OF TIMES
15         ;   SS   SINGLE STEP
16         ; *****
17         ;
18         ;   DISCLAIMER:
19         ;   IN ORDER FOR THE BREAKPOINT ROUTINES TO WORK PROPERLY
20         ;   THERE MUST BE A 'PIPELINE REGISTER ONLY' CLOCK. THIS
21         ;   WOULD ALLOW THE CHANGING OF PLR CONTENTS WITHOUT CAUSING
22         ;   AN INSTRUCTION TO BE EXECUTED. AS IT STANDS NOW, THE PIPE-
23         ;   LINE REGISTER CLOCK IS TIED TO THE SEQUENCER CLOCK. WHEN
24         ;   A BREAKPOINT HAS BEEN REACHED, ANY OPERATION INVOLVING
25         ;   A PLR CLOCK (EG. PRINTING MICROPGM MEMORY) WILL EXECUTE
26         ;   THE INSTRUCTION IN THE PLR. THE 2900 SEQUENCER STACK IS
27         ;   ALSO SOMETIMES CHANGED.
28         ; *****
29         ;
30         ;
31 000000    000000    SSCNT: .WORD 0          ;NUMBER OF TIMES TO SINGLE STEP
32 000002    000000    SSTR: .WORD 0          ;WHERE TO START SINGLE STEPPING
33         ;
34         ;   DEBUGGING ROUTINES.
35         ;
36         ;   SET BREAKPOINT
37         ;   EG. CP>AT 22
38         ;
39 000004    AT2:
40 000004        CALL FIND          ;LOOK FOR BKPT ADDR IN COMMAND LINE
41 000010    103004        BCC 1$          ;OK, CONVERT BKPT ADDRESS
42 000012        CALL ERR4          ;NOTHING THERE
43 000016    000167    000000G        JMP CPXX
44 000022        1$: CALL PACK          ;CONVERT BKPT ADDRESS
45 000026    103004        BCC 2$          ;OK, CONTINUE
46 000030        CALL ERR5          ;ERROR ON CONVERSION
47 000034    000167    000000G        JMP CPXX
48 000040    026767    000000G-000000G-2$: CMP CSHIGH,BINWD
49 000046    003004        BGT 3$          ;IS START ADDRESS IN RANGE
50 000050        CALL ERR10         ;YES, CONTINUE
51 000054    000167    000000G        JMP CPXX
52         ;
53 000060    012746    000400        3$: MOV #<CBKCLKEN>,-(SP)
54 000064        CALL CPRC          ;SET CP CNTL FOR BREAKPOINT
55 000070    016746    000000G        MOV BINWD,-(SP)
56 000074        CALL LBOP          ;LOAD DATA WORD INTO LOAD BUS REG
57 000100    052767    000000G-000000G    CALL LBCP
58         ;
59         ;   SET BCE CNTL WORD
60         ;   SET FLAG FOR BKPT SET

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

58 000106 000167 000000G..... JMP.... CPXX
59                                     ;
60                                     ;
61                                     ;
62                                     ; REMOVE BREAKPOINT (DISABLE)
63                                     ; CP>OF.
64                                     ;
65 000112.. QF2::
66 000112 042767 000000G.000000G. BIC. #BREAK,BASE. ;CLEAR BREAKPOINT FLAG.
67 000120 005046 CLR. -(SP) ;SEND 0 TO CP CNTL REG.
68 000122 CALL. CPCR ;EXECUTE TRANSFER.
69 000126 905067 176422 CLR. QR#CR2. ;CLEAR CSR2.
70 000132 000167 000000G. JMP. CPXX ;GET NEXT COMMAND.
71                                     ;
72                                     ;
73                                     ; LOAD ALL CP REGISTERS WITH THE SAME PATTERN.
74                                     ; CP>LR FFFF.
75                                     ;
76                                     ;
77 000136 LR2::
78 000136 CALL. FIND ;LOOK FOR PATTERN IN COMMAND LINE.
79 000142 103004 BCC. 1$ ;OK, CONVERT PATTERN
80 000144 CALL. ERR4 ;NOTHING THERE.
81 000150 000167 000000G. JMP. CPXX
82 000154 1$ CALL. PACK ;CONVERT PATTERN (SET UP BINUD)
83 000160 103004 BCC. 2$ ;OK, CONTINUE.
84 000162 CALL. ERR5 ;ERROR ON CONVERSION
85 000166 000167 000000G. JMP. CPXX
86                                     ;
87                                     ; CALL SUBROUTINE IN CP TO LOAD REGISTERS.
88                                     ;
89 000172 012704 000020 2$ MOV. #16,,R4 ;16 REGISTERS.
90 000176 005000 CLR. R0 ;REGISTER INDICATOR.
91 000200 3$ CALL. LDREG. ;LOAD A CP REGISTER.
92 000204 005200 INC. R0 ;INC TO NEXT REG.
93 000206 005304 DEC. R4 ;FINISHED ALL REGS?
94 000210 001373 BNE. 3$ ;NO, CONTINUE.
95 000212 000167 000000G. JMP. CPXX
96                                     ;
97                                     ;
98                                     ; PRINT CP REGISTERS.
99                                     ; CP>RG.
100                                     ; CALL SUBROUTINE IN CP TO READ REGISTERS.
101                                     ;
102                                     ;
103 000216 RG2::
104 000216 012704 000002 MOV. #2,R4 ;PRINT IN 2 SETS OF 8 REGS EACH.
105 000222 005000 CLR. R0 ;REGISTER INDICATOR.
106 000224 012705 000000G. 1$ MOV. #PRINT,R5 ;R5 -> PRINT LINE.
107 000230 012703 000010 MOV. #8,,R3 ;PRINT 8 REGS ON A LINE.
108 000234 2$ CALL. REREG. ;READ A CP REGISTER.
109 000240 CALL. UNPK ;CONVERT TO ASCII.
110 000244 005205 INC. R5 ;BUMP PRINT LINE POINTER.
111 000246 005200 INC. R0 ;BUMP REG INDICATOR.
112 000250 005303 DEC. R3
113 000252 001370 BNE. 2$ ;STAY ON CURRENT LINE.
114 000254 CALL. CONSOL. ;WRITE LINE TO CONSOLE.

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

115 000260 005304 DEC... R4
116 000262 001360 BNE... 1$ :DO SECOND SET OF 8
117 000264 000167 000000G JMP... CPXX
118 :
119 :
120 : SINGLE STEP A NUMBER OF TIMES
121 : CP>PC 4
122 :
123 :
124 000270 PC2::
125 000270 CALL... FIND :LOOK FOR COUNT IN COMMAND LINE
126 000274 103004 BCC... 1$ :OK, CONTINUE
127 000276 CALL... ERR4 :COUNT MISSING
128 000302 000167 000000G JMP... CPXX
129 000306 1$: CALL... PACK :CONVERT COUNT
130 000312 103004 BCC... 2$ :OK, CONTINUE
131 000314 CALL... ERR5 :ERROR ON CONVERSION
132 000320 000167 000000G JMP... CPXX
133 :
134 000324 005767 000000G 2$: TST... BINWD :COUNT = 0 ?
135 000330 001004 BNE... 3$ :NO, CONTINUE
136 000332 CALL... ERR5 :COUNT MUST BE GT 0
137 000336 000167 000000G JMP... CPXX
138 000342 016767 000000G 177430 3$: MOV... BINWD,SSCNT :SAVE COUNT
139 000350 000460 BR... SSCLR :ENTER SS COMMAND
140 :
141 :
142 : SINGLE STEP
143 : CP>SS
144 : CP>SS 6
145 : CP>SS 6 2
146 :
147 :
148 000352 SS2::
149 000352 CALL... FIND :LOOK FOR START ADDR IN COMMAND LINE
150 000356 103455 BCS... SSCLR :NONE THERE, USE CURRENT MAR
151 000360 CALL... PACK :CONVERT START ADDRESS
152 000364 103004 BCC... 1$ :OK, CONTINUE
153 000366 CALL... ERR5 :ERROR ON CONVERSION
154 000372 000167 000000G JMP... CPXX
155 :
156 000376 026767 000000G 000000G 1$: CMP... CSHIGH,BINWD :IS START ADDRESS IN RANGE
157 000404 003004 BGT... 2$ :YES, CONTINUE
158 000406 CALL... ERR10 :OUT OF RANGE HIGH
159 000412 000167 000000G JMP... CPXX
160 :
161 : SS START ADDRESS FOUND AND CONVERTED, NOW LOOK FOR
162 : COUNT (NUMBER OF SINGLE STEPS), EG. IN THE COMMAND
163 : CP>SS 6 2
164 : LOOK FOR THE *2*
165 :
166 000416 016767 000000G 177356 2$: MOV... BINWD,SSRT :SAVE START ADDRESS
167 000424 CALL... FIND :LOOK FOR COUNT IN COMMAND LINE
168 000430 103421 BCS... 5$ :NOTHING THERE, STOP ONLY
169 000432 CALL... PACK :CONVERT COUNT
170 000436 103004 BCC... 3$ :OK, CONTINUE
171 000440 CALL... ERR5 :ERROR ON CONVERSION

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

172 000444 000167 000000G JMP CPXX
173 ;
174 000450 005767 000000G 3$: TST BINWD ;COUNT = 0 ?
175 000454 001004 BNE 4$ ;NO. CONTINUE
176 000456 CALL ERR5 ;COUNT MUST BE GT 0
177 000462 000167 000000G JMP CPXX
178 000466 016767 000000G 177304 4$: MOV BINWD,SSCNT ;SAVE COUNT
179 ;
180 ; SEQUENCE UP TO SINGLE STEP ADDRESS (IF NECESSARY)
181 ; SINGLE CLOCK ALL TO EXECUTE ONE INSTRUCTION
182 ; NB. SINGLE STEP CANNOT BE USED IN SENDING THE
183 ; BCL ADDRESS TO THE CP.
184 ;
185 000474 016746 177302 5$: MOV SSTR,-(SP) ;PROVIDE START ADDRESS FOR SEQUENCER
186 000500 CALL SEQCS ;SEQUENCE ONLY UP TO START ADDRESS
187 000504 005046 CLR -(SP) ;CLEAR CP CONTROL REG
188 000506 CALL CPCR
189 ;
190 ; SET UP FOR SINGLE CLOCK
191 ;
192 000512 012767 100000 176422 SSCLR: MOV #Q$SM,QR$CR2 ;SET SEARCH MODE
193 000520 012746 036000 3$: MOV #<Q$CNC+Q$RNC>,-(SP) ;CLEAR PPS AND CP NO-CLOCKS
194 000524 052716 000360 BIS #<Q$CSEL>,(SP) ;CLEAR SELECT BITS
195 000530 052716 001001 BIS #<Q$LBD+Q$LBP>,(SP) ;CLEAR DRIVE AND PULSE
196 000534 012746 020000 MOV #Q$CSP,-(SP) ;SET SINGLE CLOCK ALL
197 000540 CALL CSR1 ;WRITE CSR1
198 000544 005046 CLR -(SP) ;CLEAR NOTHING
199 000546 012746 036000 MOV #<Q$CNC+Q$RNC>,-(SP) ;SET PPS AND CP NO-CLOCKS
200 000552 CALL CSR1 ;AND RE-WRITE CSR1
201 ;
202 ; SINGLE STEP A NUMBER OF TIMES DEPENDING UPON THE
203 ; COUNT IN FIELD 'SSCNT'
204 ;
205 000556 005067 176422 CLR QR$CR2 ;SET LOAD MODE
206 000562 005767 177212 TST SSCNT ;SS A NUMBER OF TIMES ?
207 000566 001403 BEQ 4$ ;NO. EXIT SS RTN
208 000570 005367 177204 DEC SSCNT
209 000574 001346 BNE SSCLR ;EXECUTE NEXT INSTRUCTION
210 ;
211 000576 4$: CALL CPMP ;PRINT MICROPGM ADDRESS
212 000602 000167 000000G JMP CPXX

```

Approved For Release 2005/07/11 : CIA-RDP85-00514R000200020001-3

ALUCKE = 040000	BYTE39 = 000047	BYTE90 = 000132	MMOE = 000004	Q\$MSET = 000004
ALUOE = 004000	BYTE4 = 000004	BYTE91 = 000133	MMJRT = 000010	Q\$MSP = 100000
AT2 = 000004RG	002 BYTE40 = 000050	BYTE92 = 000134	MNOBRE = 100000	Q\$NCLK = 176000
A01 = 010000	BYTE41 = 000051	BYTE93 = 000135	MREN1 = 000001	Q\$PP = 000100
BASE = ***** GX	BYTE42 = 000052	BYTE94 = 000136	MREN2 = 020000	Q\$PPSW = 000320
BINWD = ***** GX	BYTE43 = 000053	BYTE95 = 000137	MSYN = 000040	Q\$PP2 = 000300
BITVAL = 000000	BYTE44 = 000054	BYTE96 = 000140	N = 000144	Q\$QHLT = 000013
BIT0 = 000001	BYTE45 = 000055	BYTE97 = 000141	OF2 = 000112RG	002 Q\$QL = 000043
BIT1 = 000002	BYTE46 = 000056	BYTE98 = 000142	PACK = ***** GX	Q\$QLA = 000053
BIT10 = 000200	BYTE47 = 000057	BYTE99 = 000143	PC2 = 000270RG	002 Q\$QLB = 000054
BIT11 = 004000	BYTE48 = 000060	BYTVAL = 000144	PLB = 000010	Q\$QLR = 000001
BIT12 = 010000	BYTE49 = 000061	CE\$ALL = 001000	PLC = 000020	Q\$QW = 000042
BIT13 = 020000	BYTE5 = 000005	CBKCLK = 000400	PLD = 000030	Q\$RDCD = 000005
BIT14 = 040000	BYTE50 = 000062	CNOBRE = 100000	PLRWJ = 000200	Q\$RDMD = 000006
BIT15 = 100000	BYTE51 = 000063	CONSOL = ***** GX	PLR,EN = 000200	Q\$REBK = 001000
BIT2 = 000004	BYTE52 = 000064	CPCCEN = 010000	PRINT = ***** GX	Q\$RNC = 006000
BIT3 = 000010	BYTE53 = 000065	CPCR = ***** GX	Q\$RCR1 = 176420	Q\$RSC = 004000
BIT4 = 000020	BYTE54 = 000066	CPLB = ***** GX	Q\$RCR2 = 176422	Q\$RSET = 000010
BIT5 = 000040	BYTE55 = 000067	CPMP = 000606R	002 Q\$RLBR = 176424	Q\$SM = 100000
BIT6 = 000100	BYTE56 = 000070	CPREAD = 040000	Q\$ATTH = 000100	Q\$SP = 000120
BIT7 = 000200	BYTE57 = 000071	CPURTE = 020000	Q\$BCL = 000001	Q\$SP2 = 000340
BIT8 = 000400	BYTE58 = 000072	CPXX = ***** GX	Q\$CCCP = 000040	RERG = ***** GX
BIT9 = 001000	BYTE59 = 000073	CSADRD = 000004	Q\$CHB = 000400	RGQ,EN = 000200
BREAK = ***** GX	BYTE6 = 000006	CSEQCI = 100000	Q\$CHRL = 000200	RGQ,VA = 020000
BYTE0 = 000000	BYTE60 = 000074	CSHIGH = ***** GX	Q\$CLR = 000040	RG2 = 000216RG
BYTE1 = 000001	BYTE61 = 000075	CSOE = 000040	Q\$CNC = 030000	002 SEQCS = ***** GX
BYTE10 = 000012	BYTE62 = 000076	CSR1 = ***** GX	Q\$CP = 000060	SEQ,CI = 000010
BYTE11 = 000013	BYTE63 = 000077	CSURTE = 000100	Q\$CPC = 000010	SSCLR = 000512R
BYTE12 = 000014	BYTE64 = 000100	DBR,RP = 000001	Q\$CP2 = 000260	002 SSCNT = 000000R
BYTE13 = 000015	BYTE65 = 000101	DB\$CDD = 001457	Q\$CSC = 010000	002 SSTR = 000002R
BYTE14 = 000016	BYTE66 = 000102	DB\$SPT = 000026	Q\$CSEL = 000360	002 SS2 = 000352RG
BYTE15 = 000017	BYTE67 = 000103	DB\$TPC = 000023	Q\$CSET = 000002	S\$CLR = 000000
BYTE16 = 000020	BYTE68 = 000104	DISPGS = 100000	Q\$CSP = 020000	S\$LA = 000001
BYTE17 = 000021	BYTE69 = 000105	DMAWR = 000005	Q\$DMA = 000001	S\$QB = 000005
BYTE18 = 000022	BYTE7 = 000007	DMARRD = 000003	Q\$ENBK = 040000	S\$QR = 000006
BYTE19 = 000023	BYTE70 = 000106	DMARWR = 000004	Q\$ENOP = 020000	S\$QX = 000004
BYTE2 = 000002	BYTE71 = 000107	ENBR = 010000	Q\$FAL = 004000	S\$SR = 000007
BYTE20 = 000024	BYTE72 = 000110	ERR10 = ***** GX	Q\$FC = 000045	S\$S1 = 000010
BYTE21 = 000025	BYTE73 = 000111	ERR4 = ***** GX	Q\$FO = 000044	SSS2 = 000014
BYTE22 = 000026	BYTE74 = 000112	ERR5 = ***** GX	Q\$FP = 000046	TD\$CTR = 176370
BYTE23 = 000027	BYTE75 = 000113	FIND = ***** GX	Q\$HBF = 000002	TD\$CTU = 176360
BYTE24 = 000030	BYTE76 = 000114	LBCP = ***** GX	Q\$ICP = 000006	TD\$INL = 004000
BYTE25 = 000031	BYTE77 = 000115	LDREG = ***** GX	Q\$IH = 000003	TD\$INR = 000270
BYTE26 = 000032	BYTE78 = 000116	LOC,EN = 000100	Q\$IHRL = 000002	TD\$OAR = 176344
BYTE27 = 000033	BYTE79 = 000117	LOC,WA = 040000	Q\$IMRP = 000007	TD\$OTR = 176346
BYTE28 = 000034	BYTE8 = 000010	LOC,WB = 100000	Q\$LBD = 001000	TD\$ORD = 000274
BYTE29 = 000035	BYTE80 = 000120	LR2 = 000136RG	002 Q\$LBIP = 001001	TD\$SW = 176376
BYTE3 = 000003	BYTE81 = 000121	MAREN1 = 000001	Q\$LBP = 000001	TD\$TAR = 176372
BYTE30 = 000036	BYTE82 = 000122	MAREN2 = 040000	Q\$LCD = 000003	TD\$TAU = 176362
BYTE31 = 000037	BYTE83 = 000123	MARLOD = 010000	Q\$LDMD = 000004	Q\$LDPP = 002000
BYTE32 = 000040	BYTE84 = 000124	MAROUT = 000002	Q\$LHP = 010000	Q\$MNC = 140000
BYTE33 = 000041	BYTE85 = 000125	MAR,LO = 002000	Q\$MR = 000052	Q\$MRP = 000040
BYTE34 = 000042	BYTE86 = 000126	MAR,OU = 000040	Q\$MRP2 = 000240	Q\$MSC = 040000
BYTE35 = 000043	BYTE87 = 000127	MBKALL = 001000		
BYTE36 = 000044	BYTE88 = 000130	MBKCLK = 000400		
BYTE37 = 000045	BYTE89 = 000131	MMADDR = 000100		
BYTE38 = 000046	BYTE9 = 000011	MMLEFT = 000002		

CPBUG1: MACRO: 11110 27-MAR-80 14:41 PAGE 6-2
SYMBOL TABLE

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

T\$BTD = .002000	T\$SEG2 = .000001	WORD27 = .000066	WORD52 = .000150	WORD78 = .000234
T\$CD = .000100	T\$SEG3 = .000002	WORD28 = .000070	WORD53 = .000152	WORD79 = .000236
T\$CLK = .002000	T\$SO = .000001	WORD29 = .000072	WORD54 = .000154	WORD8 = .000020
T\$DISK = .000200	T\$UBUS = .100000	WORD3 = .000006	WORD55 = .000156	WORD80 = .000240
T\$DRD = .000004	T\$1CLK = .000400	WORD30 = .000074	WORD56 = .000160	WORD81 = .000242
T\$EMEM = .010000	T\$BBEN = .000020	WORD31 = .000076	WORD57 = .000162	WORD82 = .000244
T\$FSAA = .000000	UBD, IN = .000020	WORD32 = .000100	WORD58 = .000164	WORD83 = .000246
T\$FSAB = .000004	UNPK = .000000 GX.	WORD33 = .000102	WORD59 = .000166	WORD84 = .000250
T\$FSAC = .000014	WORD0 = .000000	WORD34 = .000104	WORD6 = .000014	WORD85 = .000252
T\$FSB2 = .000010	WORD1 = .000002	WORD35 = .000106	WORD60 = .000170	WORD86 = .000254
T\$IB = .000026	WORD10 = .000024	WORD36 = .000110	WORD61 = .000172	WORD87 = .000256
T\$IBAR = .000024	WORD11 = .000026	WORD37 = .000112	WORD62 = .000174	WORD88 = .000260
T\$IBE = .020000	WORD12 = .000030	WORD38 = .000114	WORD63 = .000176	WORD89 = .000262
T\$IBF = .040000	WORD13 = .000032	WORD39 = .000116	WORD64 = .000200	WORD9 = .000022
T\$ICD = .000040	WORD14 = .000034	WORD4 = .000010	WORD65 = .000202	WORD90 = .000264
T\$MODE = .004000	WORD15 = .000036	WORD40 = .000120	WORD66 = .000204	WORD91 = .000266
T\$OB = .000036	WORD16 = .000040	WORD41 = .000122	WORD67 = .000206	WORD92 = .000270
T\$OBE = .004000	WORD17 = .000042	WORD42 = .000124	WORD68 = .000210	WORD93 = .000272
T\$OBF = .010000	WORD18 = .000044	WORD43 = .000126	WORD69 = .000212	WORD94 = .000274
T\$OBRA = .000034	WORD19 = .000046	WORD44 = .000130	WORD7 = .000016	WORD95 = .000276
T\$OBWA = .000032	WORD2 = .000004	WORD45 = .000132	WORD70 = .000214	WORD96 = .000300
T\$OUTA = .100000	WORD20 = .000050	WORD46 = .000134	WORD71 = .000216	WORD97 = .000302
T\$RBD0 = .000200	WORD21 = .000052	WORD47 = .000136	WORD72 = .000220	WORD98 = .000304
T\$RNB = .000040	WORD22 = .000054	WORD48 = .000140	WORD73 = .000222	WORD99 = .000306
T\$RSET = .040000	WORD23 = .000056	WORD49 = .000142	WORD74 = .000224	WORDVAL = .000310
T\$SC = .000022	WORD24 = .000060	WORD5 = .000012	WORD75 = .000226	XTREAD = .001000
T\$CLK = .020000	WORD25 = .000062	WORD50 = .000144	WORD76 = .000230	XTWRITE = .000400
T\$SEG1 = .000000	WORD26 = .000064	WORD51 = .000146	WORD77 = .000232	

. ABS. 000000 000
000000 001
CPBUG1 000654 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3220 WORDS. (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS. (14 PAGES)
ELAPSED TIME: 00:00:44
CPBUG1, CPBUG1, SP=C20, 1JIM, C20, 1JCPBUG1

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

42.      ;
43.      ;
44.      ;      DEBUGGING ROUTINES
45.      ;
46.      ;
47.      ;      PROCEED
48.      ;      THE 'GO' COMMAND RUNS THE CP MICROCODE ONLY
49.      ;      TO RUN BOTH CP AND MRP USE 'FR'
50.      ;      EG. CP>GO
51.      ;      CP>GO 22
52.      ;
53. 000136      G02::
54. 000136      CALL FIND      ;LOOK FOR START ADDRESS
55. 000142 103443      BCS CGO      ;NONE, GO FROM PRESENT ADDRESS
56. 000144      CALL PACK      ;CONVERT START ADDRESS
57. 000150 103004      BCC 1$      ;OK, CONTINUE
58. 000152      CALL ERR5      ;ERROR ON CONVERSION
59. 000156 000167 000000G      JMP CPXX
60.      ;
61. 000162 026767 000000G-000000G 1$      CMP CSHIGH,BINWD      ;IS START ADDRESS IN RANGE
62. 000170 003004      BGT 2$      ;YES, CONTINUE
63. 000172      CALL ERR10      ;OUT OF RANGE HIGH
64. 000176 000167 000000G      JMP CPXX
65.      ;
66.      ;      IF THE COMMAND IS:
67.      ;      CP>GO 0
68.      ;      THEN SCAN THE COMMAND LINE FURTHER FOR AN 'N'
69.      ;      CP>GO 0 N
70.      ;      THE 'N' MEANS NOT TO ERASE THE QLB'S SO THAT THEY MAY
71.      ;      BE LOOKED AT.
72.      ;
73. 000202 005767 000000G      2$      TST BINWD      ;GO FROM ZERO
74. 000206 001015      BNE 4$      ;NO, CONTINUE
75. 000210      CALL FIND      ;LOOK FOR 'N' IN COMMAND LINE
76. 000214 103412      BCS 4$      ;NOT THERE, ERASE QLB
77. 000216 122711 000116      CMPB #'N,(R1)      ;MAKE SURE IT IS AN 'N'
78. 000222 001404      BEQ 3$      ;OK, CONTINUE
79. 000224      CALL ERR12
80. 000230 000167 000000G      JMP CPXX      ;BAD COMMAND
81. 000234 052767 000000G-000000G 3$      BIS #NEQLB,BASE      ;SET FLAG FOR NO QLB ERASE
82. 000242 016746 000000G      4$      MOV BINWD,-(SP)      ;PROVIDE START ADDRESS FOR SEQUENCER
83. 000246      CALL SEQCS      ;SEQUENCE ONLY UP TO START ADDRESS
84.      ;
85.      ;      SET UP TO FIELD INTERRUPTS
86.      ;      'OPERATIONAL INTERRUPTS' ARE HIT BUFFER INTERRUPTS, HRL
87.      ;      INTERRUPTS, AND DMA INTERRUPTS
88.      ;
89. 000252      CGO:
90. 000252 012767 000600 176422      MOV #Q$CHB+Q$CHRL>,QR$CR2      ;CLEAR OPERATIONAL INTERRUPTS
91. 000260 012767 001000 176422      MOV #Q$REBK,QR$CR2      ;RE-ARM INTERRUPTS
92.      ;
93.      ;      READ MICROPGM ADDRESS TO SEE WHERE SEQUENCER
94.      ;      IS CURRENTLY SITTING
95.      ;
96. 000266 012746 000004      MOV #CSADR,-(SP)
97. 000272      CALL CPCR      ;ENABLE CS ADDRESS - ALSO REINHIBIT BR
98. 000276      CALL CPLB      ;REQUEST CP TO LOD BUS

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

99 000302 005046          CLR - (SP)          ; CLEAR THE CONTROL REGISTER. NS
100 000304          CALL CPCR          ; NS
101 000310 012601        MOV (SP)+,R1    ; CP WORD RETURNED ON STACK
102 000312 042701 176000 BIC #176000,R1 ; CLEAR UNUSED BITS
103
104 ;
105 ;
106 000316 005701        TST R1          ; START FROM ZERO
107 000320 001011        BNE 10$         ; NO, CONTINUE
108 000322          CALL PPINIT         ; DO IT
109
110 000325 012700 000000G  MOV #HLB,R0    ; POINT TO HIT BUFFER
111 000332 016702 000000G  MOV BCL,R2    ; LOAD NUMBER OF WORDS IN HIT BUFFER
112 000336 005020 20$     CLR (R0)+      ; CLEAR HIT LIST
113 000340 005302        DEC R2
114 000342 001375        BNE 20$
115
116 ;
117 ;
118 ;
119 ;
120 000344 032767 000000G 000000G 10$     BIT #BREAK,BASE ; PROCEED FROM BREAKPOINT
121 000352 001075        BNE CBK        ; YES, SET UP FOR BREAKPOINT
122 000354 012767 120000 176422  MOV #<Q$SM+Q$ENOP>,QR$CR2 ; ENABLE OPERATIONAL INTERRUPTS
123 000362 012746 000360  MOV #<Q$CSEL>,-(SP) ; CLR SELECT BITS
124 000366 052716 001001  BIS #<Q$LBD+Q$LBP>,(SP) ; CLEAR DRIVE AND PULSE
125 000372 052716 036000  BIS #<Q$CNC+Q$RNC>,(SP) ; CLEAR PPS AND CP NO-CLOCKS
126 000376 005046        CLR -(SP)      ; SET NOTHING
127 000400          CALL CSR1           ; RE-WRITE CSR1
128
129 ;
130 ;
131 ;
132 000404 005701        TST R1          ; START FROM ZERO?
133 000406 001026        BNE 1$         ; WORD NOT ZERO
134 000410 012767 000001 176424  MOV #Q$BCL,QR$LBR ; SET ATTN CODE = 1
135 000416 012767 120100 176422  MOV #<Q$ATTN+Q$SM+Q$ENOP>,QR$CR2 ; SET ATTN CODE READY
136 000424 016701 176422  MOV QR$CR2,R1 ; READ CSR2
137 000430 032701 000100 11$     BIT #Q$ATTN,R1 ; IS ATTN BIT CLEAR
138 000434 001373        BNE 11$        ; NO, LOOP UNTIL IT IS
139
140 000436 012767 000000G 176424  MOV #BCL,QR$LBR ; MOVE BCL ADDR TO LOD BUS REG
141 000444 012767 120040 176422  MOV #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ; SET CC TO CP SLAVE DATA READY
142 000452 016701 176422 12$     MOV QR$CR2,R1 ; READ CSR2
143 000456 032701 000040 11$     BIT #Q$CCCP,R1 ; IS CC TO CP CLEAR
144 000462 001373        BNE 12$        ; NO, LOOP UNTIL IT IS
145
146 ;
147 ;
148 ;
149 ;
150 ;
151 000464 1$          CLEF$S #EFN.4      ; CLEAR QIO EVENT FLAG
152
153 000476          CALL HANG2          ; ISSUE QIO TO TERMINAL
154 000502          CALL WINT           ; WAIT FOR EVENTS
155 000506          CALL KILL           ; KILL QIO

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

156      ;
157      ;
158      ;
159 000512 005046      CLR      -(SP)      ;CLEAR NOTHING
160 000514 012746 036000 MOV      #<Q$CNC+Q$RNC>,-(SP) ;SET PPS AND CP NO-CLOCKS
161 000520      CALL     CSR1      ;WRITE CSR 1
162 000524 005046      CLR      -(SP)      ;SET CP CONTROL REG TO ZERO
163 000526      CALL     CPCR      ;DO IT
164 000532 005067 176422 CLR      QR$CR2      ;SET LOAD MODE
165 000536      CALL     CPMF      ;PRINT MICROPGM ADDRESS
166 000542 000167 000000G JMP      CPXX
167      ;
168      ;
169      ;
170      ;
171 000546      CBK:
172 000546      CLEF$S  #EFN.3      ;CLEAR BREAKPOINT EVENT FLAG
173      ;
174 000560      GLEF$S  #EFN.4      ;CLEAR QIO EVENT FLAG
175      ;
176      ;
177      ;
178      ;
179      ;
180 000572 012746 001000 MOV      #CBKALLOW,-(SP)      ;SET CP CONTROL BIT
181 000576      CALL     CPCR      ;WRITE CP CONTROL WORD
182 000602 012767 160000 176422 MOV      #<Q$SM+Q$ENBK+Q$ENOP>,QR$CR2 ;SET SEARCH MODE
183 000610 012746 000360      MOV      #<Q$CSEL>,-(SP)      ;CLR SELECT BITS
184 000614 052716 001001 BIS      #<Q$LBD+Q$LBP>,(SP)      ;CLEAR DRIVE AND PULSE
185 000620 052716 036000 BIS      #<Q$CNC+Q$RNC>,(SP)      ;CLEAR PPS AND CP NO-CLOCKS
186 000624 005046      CLR      -(SP)      ;SET NOTHING
187 000626      CALL     CSR1      ;RE-WRITE CSR1
188      ;
189      ;
190      ;
191      ;
192 000632 005701      TST      R1      ;START FROM ZERO?
193 000634 001026      BNE      1$      ;WORD NOT ZERO
194 000636 012767 000001 176424 MOV      #Q$BCL,QR$LBR      ;SET ATTN CODE = 1
195 000644 012767 120100 176422 MOV      #<Q$ATTN+Q$SM+Q$ENOP>,QR$CR2 ;SET ATTN CODE READY
196 000652 016701 176422 11$: MOV      QR$CR2,R1      ;READ CSR2
197 000656 032701 000100 BIT      #Q$ATTN,R1      ;IS ATTN BIT CLEAR
198 000662 001373      BNE      11$      ;NO, LOOP UNTIL IT IS
199      ;
200 000664 012767 000000G 176424 MOV      #BCL,QR$LBR      ;MOVE BCL ADDR TO LOD BUS REG
201 000672 012767 120040 176422 MOV      #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP SLAVE DATA READY
202 000700 016701 176422 12$: MOV      QR$CR2,R1      ;READ CSR2
203 000704 032701 000040 BIT      #Q$CCCP,R1      ;IS CC TO CP CLEAR
204 000710 001373      BNE      12$      ;NO, LOOP UNTIL IT IS
205      ;
206      ;
207      ;
208      ;
209      ;
210      ;
211 000712 1$: CALL     HANG2      ;ISSUE QIO
212 000716      CALL     WINT      ;WAIT FOR INTERRUPTS

```

213				:		
214				:	RESET. NO-CLOCKS. IN. CSR 1	
215				:	SET. LOAD. MODE.	
216				:	PRINT. MICROPGM. ADDRESS	
217				:		
218	000722.	005046			CLR. -(SP)	; CLEAR. NOTHING.
219	000724	012746	036000		MOV. #<Q\$CNC+Q\$RNC>.- (SP)	; SET. PPS. AND. CP. NO-CLOCKS.
220	000730				CALL. CSR1	; RE-WRITE. CSR1
221	000734	005046			CLR. -(SP)	; SEND. 0 TO CP. CONTROL. REG.
222	000736				CALL. CPCR	; DO. IT.
223				:		
224	000742.	005067	176422		CLR. QR\$CR2.	; SET. LOAD. MODE.
225	000746	012767	001000 176422.		MOV. #0\$REBK, QR\$CR2	; RE-ARM. INTERRUPTS.
226	000754				CALL. CPMP	; PRINT. MICROPGM. ADDRESS.
227	000760				CALL. KILL	; KILL. AST.
228	000764	000167	000000G.		JMP. CPXX	

```

230      ;
231      ;
232      ; FREE RUN
233      ; RUN BOTH MRP AND CP
234      ; EG CP>FR
235      ; CP>FR N
236      ; THE 'N' MEANS NOT TO ERASE THE QLB'S SO THAT THEY MAY
237      ; BE LOOKED AT.
238      ;
239      ;
240      000770      FR2::
241      000770      CALL    FIND          ;LOOK FOR 'N' IN COMMAND LINE
242      000774      103412      BCS      20$          ;NOT THERE, ERASE QLB
243      000776      122711      000116      CMPB    #'N',(R1)      ;MAKE SURE IT IS AN 'N'
244      001002      001404      BEQ      10$          ;OK, CONTINUE
245      001004      CALL    ERR12
246      001010      000167      000000G      JMP     CPXX          ;BAD COMMAND
247      001014      052767      000000G      10$      BIS     #NEQLB,BASE ;SET FLAG FOR NO QLB ERASE
248      ;
249      ;
250      ; LOAD PPS MASK REGISTER
251      ; DO A MASTER RESET
252      ; INITIALIZE PPS
253      001022      012746      000052      20$      MOV     #Q$MR,-(SP)      ;SELECT PPS MASK REGISTER
254      001026      CALL    PPCP
255      001032      012746      037774      MOV     #037774,-(SP)      ;WRITE X'3FFC'
256      001036      CALL    LBPSC
257      ;
258      001042      012746      177777      MOV     #177777,-(SP)      ;CLEAR CSR1
259      001046      012746      000010      MOV     #Q$RSET,-(SP)      ;SET RESET
260      001052      CALL    CSR1          ;RESET HQR
261      001056      012746      000010      MOV     #Q$RSET,-(SP)      ;CLEAR RESET
262      001062      012746      176000      MOV     #<Q$MNC+Q$CNC+Q$RNC>,-(SP) ;SET NO CLKS
263      001066      CALL    CSR1          ;MOVE TO CSR1
264      ;
265      001072      CALL    PPINIT        ;INITIALIZE PPS
266      ;
267      ;
268      ; SET MRP AND CP START ADDRESSES TO ZERO
269      001076      005046      CLR     -(SP)          ;START MRP AT ZERO
270      001100      CALL    SEQMM
271      ;
272      001104      005046      CLR     -(SP)          ;REINHIBIT BRANCH CONTROL REGISTER
273      001106      CALL    MRPCR
274      ;
275      001112      005046      CLR     -(SP)          ;START CP AT ZERO
276      001114      CALL    SEQCS
277      ;
278      001120      005046      CLR     -(SP)          ;REINHIBIT BRANCH CONTROL REGISTER
279      001122      CALL    CPCR
280      ;
281      ;
282      ; SET UP FOR OPERATIONAL INTERRUPTS
283      ; OPERATIONAL INTERRUPTS ARE HIT BUFFER INTERRUPTS
284      ; HRL INTERRUPTS, AND DMA INTERRUPTS
285      ; CLEAR HIT BUFFER
286      001126      012767      000600      176422      MOV     #<Q$CHB+Q$CHRL>,(R2) ;CLEAR OPERATIONAL INTERRUPTS

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

287 001134 012767 001000 176422: MOV. #Q$REBK,QR$CR2 ;RE-ARM INTERRUPTS.
288 ;
289 001142 012700 000000G: MOV. #HLB,R0 ;POINT TO HIT BUFFER
290 001146 016701 000000G: MOV. BCL,R1 ;LOAD NUMBER OF WORDS IN HIT BUFFER
291 001152 005020 1$: CLR. (R0)+ ;CLEAR HIT LIST
292 001154 005301 DEC. R1
293 001156 001375 BNE. 1$
294 ;
295 ;
296 ; IF A BREAKPOINT IS IN EFFECT, BRANCH TO CODE THAT PREPARES
297 ; TO FIELD A BREAKPOINT INTERRUPT. ELSE START FREE RUN WITHOUT
298 ; BREAKPOINTS.
299 001160 032767 000000G-000000G: BIT. #BREAK,BASE ;ANY BREAKPOINTS SET
300 001166 001073 BNE. CBK2 ;YES, SET UP FOR BREAKPOINT
301 001170 012767 120000 176422: MOV. *(Q$SM+Q$ENOP),QR$CR2 ;ENABLE OPERATIONAL INTERRUPTS
302 001176 012746 000360 MOV. *(Q$CSEL),-(SP) ;CLR SELECT BITS
303 001202 052716 001001 BIS. *(Q$LBD+Q$LBP), (SP) ;CLEAR DRIVE AND PULSE
304 001206 052716 176000 BIS. *(Q$NCLK), (SP) ;CLEAR NO-CLOCKS
305 001212 005046 CLR. -(SP) ;SET NOTHING
306 001214 CALL. CSR1 ;RE-WRITE CSR1
307 ;
308 ;
309 ; SEND ATTENTION CODE AND BCL ADDRESS
310 001220 012767 000001 176424 MOV. #Q$BCL,QR$LBR ;SET ATTN CODE = 1
311 001226 012767 120100 176422: MOV. *(Q$ATTN+Q$SM+Q$ENOP),QR$CR2 ;SET ATTN CODE READY
312 001234 016701 176422 11$: MOV. QR$CR2,R1 ;READ CSR2
313 001240 032701 000100 BIT. #Q$ATTN,R1 ;IS ATTN BIT CLEAR
314 001244 001373 BNE. 11$ ;NO, LOOP UNTIL CLEAR
315 ;
316 001246 012767 000000G-176424 MOV. #BCL,QR$LBR ;MOVE BCL ADDR TO LOD BUS REG
317 001254 012767 120040 176422: MOV. *(Q$CCCP+Q$SM+Q$ENOP),QR$CR2 ;SET CC TO CP SLAVE DATA READY
318 001262 016701 176422 12$: MOV. QR$CR2,R1 ;READ CSR2
319 001266 032701 000040 BIT. #Q$CCCP,R1 ;IS CC TO CP CLEAR
320 001272 001373 BNE. 12$ ;NO, LOOP UNTIL CLEAR
321 ;
322 ;
323 ; PROVIDE FOR INTERRUPT FROM TERMINAL TO RETURN CONTROL
324 ; TO USER (IN CASE MICROCODE HANGS UP OR RUNS FOREVER.)
325 ; CALL SUBROUTINE THAT WAITS FOR INTERRUPTS THEN DECODES
326 ; THEM.
327 001274 CLEF$. #EFN,4 ;CLEAR QIO EVENT FLAG
328 ;
329 001306 CALL. HANG2 ;ISSUE QIO TO TERMINAL
330 001312 CALL. WINT ;WAIT FOR EVENTS
331 001316 CALL. KILL ;KILL QIO
332 ;
333 ;
334 ; TURN OFF CLOCKS, SET LOAD MODE, PRINT MICROPGM ADDRESS
335 001322 005046 CLR. -(SP) ;CLEAR NOTHING
336 001324 012746 176000 MOV. *(Q$NCLK),-(SP) ;SET NO-CLOCKS
337 001330 CALL. CSR1 ;WRITE CSR 1
338 001334 005046 CLR. -(SP) ;SET CP CONTROL REG TO ZERO
339 001336 CALL. CPCR ;DO IT
340 001342 005067 176422 CLR. QR$CR2 ;SET LOAD MODE
341 001346 CALL. CPMP ;READ CP MICROPGM ADDRESS
342 001352 000167 000000G: JMP. CPXX
343 ;

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
344      ;
345      ; EXPECT TO HIT A BREAKPOINT.
346      ;
347      CBK2:
348      CLEF$S  #EFN.3      ;CLEAR BREAKPOINT EVENT FLAG
349      ;
350      CLEF$S  #EFN.4      ;CLEAR QIO EVENT FLAG
351      ;
352      ; SET BKPT ALLOW BIT IN CP CONTROL REG.
353      ; SET SEARCH MODE
354      ; SET FREE RUN CLOCK
355      ;
356      001402 012746 001000      MOV.  #CBKALLOW, -(SP)      ;SET CP CONTROL BIT
357      001406      CPCLR      ;WRITE CP CONTROL WORD
358      001412 012767 160000 176422 MOV.  #<Q$SM+Q$ENBK+Q$ENOP>, QR$CR2 ;SET SEARCH MODE
359      001420 012746 000360      MOV.  #<Q$CSEL>, -(SP)      ;CLR SELECT BITS FW
360      001424 052716 001001      BIS.  #<Q$LBD+Q$LBP>, (SP)      ;CLEAR DRIVE AND PULSE FW
361      001430 052716 176000      BIS.  #<Q$NCLK>, (SP)      ;CLEAR NO-CLOCKS
362      001434 005046      CLR.  -(SP)      ;SET NOTHING FW
363      001436      CALL.  CSR1      ;RE-WRITE CSR1 FW
364      ;
365      ; SEND BCL ADDRESS TO CP.
366      ;
367      001442 012767 000001 176424 MOV.  #Q$BCL, QR$LBR      ;SET ATTN CODE = 1
368      001450 012767 160100 176422 MOV.  #<Q$ATTN+Q$SM+Q$ENBK+Q$ENOP>, QR$CR2 ;SET ATTN CODE READY
369      001456 016701 176422 11$: MOV.  QR$CR2, R1      ;READ CSR2
370      001462 032701 000100      BIT.  #Q$ATTN, R1      ;IS ATTN BIT CLEAR
371      001466 001373      BNE.  11$      ;NO, LOOP UNTIL CLEAR
372      ;
373      001470 012767 000000G 176424 MOV.  #BCL, QR$LBR      ;MOVE BCL ADDR TO LOD BUS REG
374      001476 012767 160040 176422 MOV.  #<Q$CCCP+Q$SM+Q$ENBK+Q$ENOP>, QR$CR2 ;SET CC TO CP SLAVE DATA READY
375      001504 016701 176422 12$: MOV.  QR$CR2, R1      ;READ CSR2
376      001510 032701 000040      BIT.  #Q$CCCP, R1      ;IS CC TO CP CLEAR
377      001514 001373      BNE.  12$      ;NO, LOOP UNTIL CLEAR
378      ;
379      ; PROVIDE FOR INTERRUPT FROM TERMINAL TO RETURN CONTROL
380      ; TO USER (IN CASE MICROCODE HANGS UP OR RUNS FOREVER)
381      ; CALL SUBROUTINE THAT WAITS FOR INTERRUPTS THEN DECODES
382      ; THEM
383      ;
384      001516      CALL.  HANG2      ;KILL QIO
385      001522      CALL.  WINT      ;WAIT FOR INTERRUPTS
386      ;
387      ; RESET NO-CLOCKS IN CSR 1
388      ; SET LOAD MODE
389      ; PRINT MICROPGM ADDRESS
390      ;
391      001526 005046      CLR.  -(SP)      ;CLEAR NOTHING
392      001530 012746 176000      MOV.  #<Q$NCLK>, -(SP)      ;SET NO-CLOCKS
393      001534      CALL.  CSR1      ;RE-WRITE CSR1
394      001540 005046      CLR.  -(SP)      ;SEND 0 TO CP CONTROL REG
395      001542      CALL.  CPCLR      ;DO IT
396      ;
397      001546 005067 176422      CLR.  QR$CR2      ;SET LOAD MODE
398      001552 012767 001000 176422 MOV.  #Q$REBK, QR$CR2      ;RE-ARM INTERRUPTS
399      001560      CALL.  CPMP      ;PRINT MICROPGM ADDRESS
400      001564      CALL.  KILL      ;KILL AST
```


CPBUG2: MACRO: M1110 27-MAR-88 14:42: PAGE 7-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

401 001570 000167 000000G.....

JMP..... CPXX

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

403      ;
404      ;
405      ; WAIT FOR INTERRUPTS.
406      ; NB. DO NOT USE 'WTSE$' OR 'WTLO$'. THEY CAUSE SYSTEM PROBLEMS.
407      ; IN CONNECTION WITH THE WAY INTERRUPTS ARE HANDLED. (SEE THE
408      ; 'BPTISR' SUBROUTINE IN QMAIN).
409      ;
410      ;
411 001574 WINT:
412 001574 100$: RDAF$S. #EFBUF. ; READ EVENT FLAGS.
413      ;
414 001606 032767 000014 000000G. BIT. *(<BIT3+BIT2>).EFBUF. ; ANY INTERRUPTS.
415 001614 001767 BEQ. 100$ ; NO. READ AGAIN.
416      ;
417      ; INTERRUPT DETECTED. DECODE IT.
418      ; NB. EVENT FLAGS NUMBERS BEGIN WITH 1. BIT NUMBERS IN THE
419      ; EVENT FLAG BUFFER BEGIN WITH 0.
420      ; IF THE INTERRUPT IS FROM THE TERMINAL, IMMEDIATELY
421      ; EXIT THIS SUBROUTINE.
422      ;
423 001616 032767 000010 000000G. BIT. #BIT3.EFBUF. ; EF #4 = INTERRUPT FROM TERMINAL.
424 001624 001402 BEQ. 10$ ; NO.
425 001626 000167 000420 JMP. WINTX2. ; EXIT IF TRUE.
426      ;
427      ; READ CSR2. CSR2 INCLUDES THE INTERRUPT FLAGS.
428      ;
429 001632 016767 176422 000000G. 10$: MOV. QR$CR2,APLACE. ; READ CSR #2.
430 001640 042767 177770 000000G. BIC. #177770,APLACE ; MASK OFF INTERRUPT BITS.
431      ;
432      ;
433      ; CHECK FOR CP BREAKPOINT INTERRUPT.
434      ; BECAUSE BREAKPOINT INTERRUPTS ARE OF THE HIGHEST PRIORITY,
435      ; CHECK TO SEE WHETHER A HIT LIST OR HRL INTERRUPT IS PENDING.
436      ; THIS WILL ONLY HAPPEN WHEN IN THE MICROCODE A BREAKPOINT
437      ; HALT IMMEDIATELY FOLLOWS AN HL OR HRL INTERRUPT ('CPBUG2.
438      ; WILL SEE THEM IN REVERSE ORDER).
439      ;
440      ; DMA INTERRUPTS WILL GO UNREPORTED UNDER THESE CIRCUMSTANCES.
441      ;
442      ;
443      ; TELL WHETHER A HIT BUFFER INTERRUPT IS PENDING BY LOOKING
444      ; AT THE BCL FOR A CLEARED ENTRY.
445      ;
446 001646 122767 000006 000000G. CMPB. #Q$ICP,APLACE. ; CP BREAKPOINT INTERRUPT.
447 001654 001000 BNE. OTHERS. ; NO. CHECK OTHERS.
448 001656 005767 000002G. TST. BCL+2. ; HIT BUFFER BCL ENTRY CLEARED.
449 001662 001025 BNE. 2$. ; NO. TEST HRL.
450 001664 012767 101000 176422. MOV. *(<Q$SM+Q$REBK>),QR$CR2. ; RE-ARM INTERRUPTS.
451 001672 012767 120000 176422. MOV. *(<Q$SM+Q$ENOP>),QR$CR2. ; ENABLE OPERATIONAL INTERRUPTS.
452 001700 012767 100600 176422. MOV. *(<Q$SM+Q$CHB+Q$CHRL>),QR$CR2. ; CLEAR OPERATIONAL INTERRUPTS.
453 001706 012767 000000G. 000002G. MOV. #HLB,BCL+2. ; RESTORE BCL ENTRY.
454 001714 012700 000017. MOV. #HBMSG,R0. ; POINT TO HIT LIST MESSAGE.
455 001720 012705 000000G. MOV. #PRINT,R5. ; POINT TO PRINT LINE.
456 001724 112025 1$. MOV. (R0)+,(R5)+
457 001726 001376 BNE. 1$.
458 001730 CALL. CONSOL. ; PRINT MESSAGE.
459 001734 000541 BR. WINTX. ; LEAVE INTERRUPT ROUTINE.

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

460      ;
461      ;
462      ; CHECK WHETHER AN HRL ENTRY IS PENDING BY LOOKING
463      ; FOR A CLEARED BCL ENTRY.
464      001736 005767 000012G 2$: TST BCL+10. ;HRL BCL ENTRY CLEARED.
465      001742 001136 BNE WINTX. ;NO EXIT.
466      001744 012767 101000 176422 MOV *(<Q$SM+Q$REBK>),Q$CR2. ;RE-ARM INTERRUPTS.
467      001752 012767 120000 176422 MOV *(<Q$SM+Q$ENOP>),Q$CR2. ;ENABLE OPERATIONAL INTERRUPTS
468      001760 012767 100600 176422 MOV *(<Q$SM+Q$CHB+Q$CHRL>),Q$CR2. ;CLEAR OPERATIONAL INTERRUPTS.
469      001766 012767 000000G 000012G MOV *HRL,BCL+10. ;RESTORE BCL ENTRY.
470      001774 012700 000042 MOV *HRLMSG,R0. ;POINT TO HIT LIST MESSAGE.
471      002000 012705 000000G MOV *PRINT,R5. ;POINT TO PRINT LINE
472      002004 112025 3$: MOVB (R0)+,(R5)+
473      002006 001376 BNE 3$
474      002010 CALL CONSOL. ;PRINT MESSAGE.
475      002014 000511 BR WINTX. ;LEAVE INTERRUPT ROUTINE.
476      ;
477      ;
478      ; CHECK FOR MRP INTERRUPT.
479      ;
480      002016 122767 000007 000000G OTHERS: CMPB *Q$IMRP,APLACE ;MRP BREAKPOINT ?
481      002024 001003 BNE 1$ ;NO TRY NEXT.
482      002026 012700 000000 MOV *MRPMSG,R0 ;R0 -> MESSAGE.
483      002032 000443 BR PMSG ;PRINT MESSAGE.
484      ;
485      ;
486      ; CHECK FOR HIT BUFFER INTERRUPT.
487      002034 122767 000003 000000G 1$: CMPB *Q$IHBA,APLACE ;HIT LIST INTERRUPT.
488      002042 001011 BNE 2$ ;NO TRY NEXT.
489      002044 012767 100400 176422 MOV *(<Q$SM+Q$CHB>),Q$CR2. ;CLEAR HL INTERRUPT.
490      002052 012767 000000G 000002G MOV *HLB,BCL+2. ;PUT BUFFER ADDRESS BACK IN BCL.
491      002060 012700 000017 MOV *HBMMSG,R0 ;POINT TO HIT LIST MESSAGE.
492      002064 000426 BR PMSG ;PRINT MESSAGE.
493      ;
494      ;
495      ; CHECK FOR HRL INTERRUPT.
496      002066 122767 000002 000000G 2$: CMPB *Q$IHRL,APLACE ;HRL INTERRUPT
497      002074 001011 BNE 3$ ;NO TRY NEXT.
498      002076 012767 100200 176422 MOV *(<Q$SM+Q$CHRL>),Q$CR2. ;CLEAR HRL INTERRUPT
499      002104 012767 000000G 000012G MOV *HRL,BCL+10. ;PUT BUFFER ADDRESS BACK IN BCL.
500      002112 012700 000042 MOV *HRLMSG,R0 ;POINT TO HRL MESSAGE.
501      002116 000411 BR PMSG
502      ;
503      ;
504      ; CHECK FOR DMA INTERRUPT.
505      ; NB. DMA USES HIT BUFFER INTERRUPT REQUEST FROM CP.
506      ; (SEE CP MICROCODE).
507      002120 122767 000001 000000G 3$: CMPB *Q$DMA,APLACE ;DMA ?
508      002126 001030 BNE UNID
509      002130 012767 100400 176422 MOV *(<Q$SM+Q$CHB>),Q$CR2. ;CLEAR DMA INTERRUPT (USING HIT BUF FLAG)
510      002136 012700 000060 MOV *DMAMSG,R0 ;POINT TO DMA MESSAGE.
511      ;
512      ;
513      ; PRINT MESSAGE. CLEAR INTERRUPT FLAG. WAIT FOR NEXT INTERRUPT.
514      002142 012705 000000G PMSG: MOV *PRINT,R5 ;POINT TO PRINT LINE
515      002146 112025 3$: MOVB (R0)+,(R5)+
516      002150 001376 BNE 3$

```

```
517 002152.          CALL  CONSOL.          ;WRITE MESSAGE TO TERMINAL.
518                  ;
519 002156          CLEF$S. #EFN.3          ;CLEAR HQR INTERRUPT EVENT FLAG.
520                  ;
521 002170 012767 101000 176422.          MOV.  #<Q$SM+Q$REBK>,QR$CR2.  ;RE-ARM INTERRUPTS.
522 002176 012767 160000 176422.          MOV.  #<Q$SM+Q$ENBK+Q$ENOP>,QR$CR2. ;ENABLE INTERRUPTS.
523 002204 000167 177364          JMP.  WINT          ;WAIT FOR NEXT INTERRUPT.
524                  ;
525                  ; UNIDENTIFIED INTERRUPT
526                  ;
527 002210 012700 000076.          UNID: MOV.  #UNMSG,R0          ;POINT TO MESSAGE.
528 002214 012705 000000G.          MOV.  #PRINT,R5          ;POINT TO PRINT LINE
529 002220 112025          5$: MOV.  (R0)+,(R5)+
530 002222 001376          BNE.  5$
531 002224 016701 000000G.          MOV.  APLACE,R1          ;LOAD CSR2 INTO R1
532 002230          CALL.  UNPK          ;CONVERT TO ASCII-HEX.
533 002234          CALL.  CONSOL.          ;WRITE MESSAGE TO TERMINAL.
534                  ;
535                  ; EXIT.
536                  ;
537 002240          WINTX: CLEF$S. #EFN.3          ;CLEAR HQR INTERRUPT EVENT FLAG.
538 002240
539                  ;
540 002252          WINTX2: RETURN.
```

```
542.
543.
544.
545.
546.
547 002254
548 002254 012746 000004
549 002260
550 002264
551 002270 012601
552 002272 042701 176000
553 002276 012705 000000G
554 002302
555 002306
556 002312 005046
557 002314
558 002320
```

PRINT: CP: MICROPROGRAM: ADDRESS:

CPMP:

MOV.	*(CSADR),-(SP)	:READ MPA SELECT BIT
CALL.	CPCR	:WRITE CP CONTROL REGISTER
CALL.	CPLB	:REQUEST CP TO LOD BUS
MOV.	(SP)+,R1	:UNLOAD MPA
BIC.	*176000,R1	:CLEAR UPPER 6 BITS
MOV.	*PRINT,R5	:POINT TO PRINT LINE
CALL.	UNPK	:CONVERT MPA
CALL.	CONSOL	:WRITE TO CONSOLE
CLR.	-(SP)	:CLEAR CP CR
CALL.	CPCR	
RETURN.		

```

560      ;
561      ;
562      ;      INITIALIZE PPS.
563      ;
564      ;
565      ;      SET HP QUIESCENT.
566      ;
567      PPINIT:
568      002322 012746 000053      MOV.    #0$OLA, -(SP)      ; ADDRESS SELECT FOR QLB PAGES.
569      002326      CALL.    PPCR
570      002332 012746 002000      MOV.    #2000, -(SP)      ; SEND ADDRESS 'X' 400' (ILLEGAL)
571      002336      CALL.    LBPP
572      ;
573      ;
574      ;      RESET MRP AND CP.
575      ;
576      002342 005046      CLR.    -(SP)      ; CLEAR NOTHING IN CSR1
577      002344 012746 000004      MOV.    #0$MSET, -(SP)      ; SET RESET
578      002350      CALL.    CSR1
579      002354 012746 000004      MOV.    #0$MSET, -(SP)      ; CLEAR RESET
580      002360 005046      CLR.    -(SP)      ; SET NOTHING
581      002362      CALL.    CSR1
582      ;
583      002366 005046      CLR.    -(SP)      ; CLEAR NOTHING IN CSR1
584      002370 012746 000002      MOV.    #0$CSET, -(SP)      ; SET RESET
585      002374      CALL.    CSR1
586      002400 012746 000002      MOV.    #0$CSET, -(SP)      ; CLEAR RESET
587      002404 005046      CLR.    -(SP)      ; SET NOTHING
588      002406      CALL.    CSR1
589      ;
590      ;      SET 'QLB ERASE'
591      ;
592      002412 032767 000000G-000000G      BIT.    #NEQLB, BASE      ; NO ERASE FOR QLB ?
593      002420 001406      BEQ.    1$      ; ERASE QLB
594      002422 042767 000000G-000000G      BIC.    #NEQLB, BASE      ; CLEAR NO ERASE FLAG
595      002430 012746 000300      MOV.    #300, -(SP)      ; SET CODE = 'X' C0
596      002434 000402      BR      2$
597      002436 012746 000100      1$:      MOV.    #100, -(SP)      ; SEND 'X' 40' TO PPCR
598      002442      CALL.    PPCR
599      002446      RETURN
600      ;
601      000001      .END

```

ALUCKE = 040000	BYTE38 = 000046	BYTE9 = 000011	LBPP = ***** GX	Q\$FC = 000045
ALUOE = 004000	BYTE39 = 000047	BYTE90 = 000132	LBPSC = ***** GX	Q\$FO = 000044
APLACE = ***** GX	BYTE4 = 000004	BYTE91 = 000133	LOC.EN = 000100	Q\$FP = 000046
A01 = 010000	BYTE40 = 000050	BYTE92 = 000134	LOC.WA = 040000	Q\$HBF = 000002
BASE = ***** GX	BYTE41 = 000051	BYTE93 = 000135	LOC.WB = 100000	Q\$ICP = 000006
BCL = ***** GX	BYTE42 = 000052	BYTE94 = 000136	MAREN1 = 000001	Q\$IHB = 000003
BINWD = ***** GX	BYTE43 = 000053	BYTE95 = 000137	MAREN2 = 004000	Q\$IHRL = 000002
BITVAL = 000000	BYTE44 = 000054	BYTE96 = 000140	MARLOD = 010000	Q\$IMRP = 000007
BIT0 = 000001	BYTE45 = 000055	BYTE97 = 000141	MAROUT = 000002	Q\$LBD = 001000
BIT1 = 000002	BYTE46 = 000056	BYTE98 = 000142	MAR.LO = 002000	Q\$LBDP = 001001
BIT10 = 002000	BYTE47 = 000057	BYTE99 = 000143	MAR.OU = 000040	Q\$LBP = 000001
BIT11 = 004000	BYTE48 = 000060	BYTVAL = 000144	MBKALL = 001000	Q\$LCD = 000003
BIT12 = 010000	BYTE49 = 000061	CBK = 000546R	002.MBKCLK = 000400	Q\$LMD = 000004
BIT13 = 020000	BYTE5 = 000005	CBKALL = 001000	MMADRD = 000100	Q\$LDPP = 002000
BIT14 = 040000	BYTE54 = 000062	CBKCLK = 000400	MMLEFT = 000002	Q\$LHP = 010000
BIT15 = 100000	BYTE51 = 000063	CBK2 = 001356R	002.MMOE = 000004	Q\$MNC = 140000
BIT2 = 000004	BYTE52 = 000064	CGO = 000252R	002.MMURTE = 000010	Q\$MR = 000052
BIT3 = 000010	BYTE53 = 000065	CNOBRE = 100000	MNOBRE = 100000	Q\$MRP = 000040
BIT4 = 000020	BYTE54 = 000066	CONSOL = ***** GX	MREN1 = 000001	Q\$MRP2 = 000240
BIT5 = 000040	BYTE55 = 000067	CPCCEN = 010000	MREN2 = 020000	Q\$MSE = 040000
BIT6 = 000100	BYTE56 = 000070	CPCR = ***** GX	MRPCR = ***** GX	Q\$MSET = 000004
BIT7 = 000200	BYTE57 = 000071	CPLB = ***** GX	MRPMSG = 000000R	002.Q\$MSP = 100000
BIT8 = 000400	BYTE58 = 000072	CPMP = 002254R	002.MSYN = 000040	Q\$NCLK = 176000
BIT9 = 001000	BYTE59 = 000073	CPREAD = 040000	N = 000144	Q\$PP = 000100
BREAK = ***** GX	BYTE6 = 000006	CPWRITE = 020000	NEQLB = ***** GX	Q\$PPSW = 000320
BYTE0 = 000000	BYTE60 = 000074	CPXX = ***** GX	OTHERS = 002016R	002.Q\$PP2 = 000300
BYTE1 = 000001	BYTE61 = 000075	CSADRD = 000004	PACK = ***** GX	Q\$QHLT = 000013
BYTE10 = 000012	BYTE62 = 000076	CSEOC1 = 100000	PLB = 000010	Q\$QL = 000043
BYTE11 = 000013	BYTE63 = 000077	CSHIGH = ***** GX	PLC = 000020	Q\$QLA = 000053
BYTE12 = 000014	BYTE64 = 000100	C\$OE = 000040	PLD = 000030	Q\$QLB = 000054
BYTE13 = 000015	BYTE65 = 000101	CSR1 = ***** GX	PLWR = 000200	Q\$QLR = 000001
BYTE14 = 000016	BYTE66 = 000102	CSWRTE = 000100	PLR.EN = 000200	Q\$QW = 000042
BYTE15 = 000017	BYTE67 = 000103	DBR.RD = 000001	PMSG = 002142R	002.Q\$RDCD = 000005
BYTE16 = 000020	BYTE68 = 000104	DB\$CPP = 001457	PPCR = ***** GX	Q\$RDM = 000006
BYTE17 = 000021	BYTE69 = 000105	DB\$SPT = 000026	PPINIT = 002322R	002.Q\$REBK = 001000
BYTE18 = 000022	BYTE7 = 000007	DB\$TPC = 000023	PRINT = ***** GX	Q\$RNC = 006000
BYTE19 = 000023	BYTE70 = 000106	DISPGS = 100000	QR\$CR1 = 176420	Q\$RSC = 004000
BYTE2 = 000002	BYTE71 = 000107	DMAUR = 000005	QR\$CR2 = 176422	Q\$RSET = 000010
BYTE20 = 000024	BYTE72 = 000110	DMMSG = 000060R	002.Q\$RLBR = 176424	Q\$SM = 100000
BYTE21 = 000025	BYTE73 = 000111	DMARRD = 000003	Q\$ATTN = 000100	Q\$SP = 000120
BYTE22 = 000026	BYTE74 = 000112	DMARUR = 000004	Q\$BCL = 000001	Q\$SP2 = 000340
BYTE23 = 000027	BYTE75 = 000113	EFBUF = ***** GX	Q\$CCCP = 000040	RGQ.EN = 000200
BYTE24 = 000030	BYTE76 = 000114	EFN.3 = ***** GX	Q\$CHB = 000400	RGQ.VA = 020000
BYTE25 = 000031	BYTE77 = 000115	EFN.4 = ***** GX	Q\$CHRL = 000200	SEQCS = ***** GX
BYTE26 = 000032	BYTE78 = 000116	ENBR = 010000	Q\$CLR = 000040	SEQM1 = ***** GX
BYTE27 = 000033	BYTE79 = 000117	ERR10 = ***** GX	Q\$CHC = 030000	SEQ.CI = 000010
BYTE28 = 000034	BYTE8 = 000010	ERR12 = ***** GX	Q\$CPC = 000060	S\$CLR = 000000
BYTE29 = 000035	BYTE80 = 000120	ERR5 = ***** GX	Q\$CPCC = 000010	S\$LA = 000001
BYTE3 = 000003	BYTE81 = 000121	FIND = ***** GX	Q\$CP2 = 000260	S\$OB = 000005
BYTE30 = 000036	BYTE82 = 000122	FR2 = 000770RG	002.Q\$CSC = 010000	S\$OR = 000006
BYTE31 = 000037	BYTE83 = 000123	GOL = 000136RG	002.Q\$CSEL = 000360	S\$OX = 000004
BYTE32 = 000040	BYTE84 = 000124	HANG2 = ***** GX	Q\$CSET = 000002	S\$SR = 000007
BYTE33 = 000041	BYTE85 = 000125	HBMSG = 000017R	002.Q\$CSP = 020000	S\$SI = 000010
BYTE34 = 000042	BYTE86 = 000126	HLB = ***** GX	Q\$ENBK = 040000	Q\$S2 = 000014
BYTE35 = 000043	BYTE87 = 000127	HRL = ***** GX	002.Q\$ENOP = 020000	TD\$CTR = 176370
BYTE36 = 000044	BYTE88 = 000130	HRLMSG = 000042R	Q\$FOL = 004000	TD\$CTW = 176360
BYTE37 = 000045	BYTE89 = 000131	KILL = ***** GX	Q\$FOL = 004000	TD\$INL = 004000

CPBUG2: M000 M1110 27-MAR-80 14:42 PAGE 10-2
SYMBOL TABLE

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

TD\$MEM= 000270	T\$OBE= 004000	WORD16= 000040	WORD45= 000132	WORD74= 000224
TD\$OAR= 176344	T\$OBF= 010000	WORD17= 000042	WORD46= 000134	WORD75= 000226
TD\$OTR= 176346	T\$OBRA= 000034	WORD18= 000044	WORD47= 000136	WORD76= 000230
TD\$QRD= 000274	T\$OBWA= 000032	WORD19= 000046	WORD48= 000140	WORD77= 000232
TD\$SW= 176376	T\$OUTA= 100000	WORD20= 000004	WORD49= 000142	WORD78= 000234
TD\$TAR= 176372	T\$RBD0= 000200	WORD21= 000050	WORD50= 000144	WORD79= 000236
TD\$TAW= 176362	T\$RNB= 000040	WORD22= 000052	WORD51= 000146	WORD80= 000240
TD\$TDR= 176374	T\$RSET= 040000	WORD23= 000056	WORD52= 000150	WORD81= 000242
TD\$TDW= 176364	T\$SC= 000022	WORD24= 000060	WORD53= 000152	WORD82= 000244
T\$AD= 000020	T\$SCLK= 020000	WORD25= 000062	WORD54= 000154	WORD83= 000246
T\$BA= 000002	T\$SEG1= 000000	WORD26= 000064	WORD55= 000156	WORD84= 000250
T\$BD= 000010	T\$SEG2= 000001	WORD27= 000066	WORD56= 000160	WORD85= 000252
T\$BS0= 100000	T\$SEG3= 000002	WORD28= 000070	WORD57= 000162	WORD86= 000254
T\$BT= 000020	T\$S0= 000001	WORD29= 000072	WORD58= 000164	WORD87= 000256
T\$BTAR= 000030	T\$UBUS= 100000	WORD30= 000074	WORD59= 000166	WORD88= 000260
T\$BTD= 002000	T\$1CLK= 000400	WORD31= 000076	WORD60= 000170	WORD89= 000262
T\$CD= 000100	T\$BBEN= 000020	WORD32= 000100	WORD61= 000172	WORD90= 000264
T\$CLK= 002000	UBD, IN= 000020	WORD33= 000102	WORD62= 000174	WORD91= 000266
T\$DISK= 000200	UNID= 002210R	WORD34= 000104	WORD63= 000176	WORD92= 000270
T\$DRD= 000004	UNMSG= 000076R	WORD35= 000106	WORD64= 000200	WORD93= 000272
T\$EMEM= 010000	UNPK= ***** GX	WORD36= 000110	WORD65= 000202	WORD94= 000274
T\$FSAA= 000000	WINT= 001574R	WORD37= 000112	WORD66= 000204	WORD95= 000276
T\$FSAB= 000004	WINTX= 002240R	WORD38= 000114	WORD67= 000206	WORD96= 000300
T\$FSAC= 000014	WINTX2= 002252R	WORD39= 000116	WORD68= 000210	WORD97= 000302
T\$FSB2= 000010	WORD0= 000000	WORD40= 000120	WORD69= 000212	WORD98= 000304
T\$IB= 000026	WORD1= 000002	WORD41= 000122	WORD70= 000214	WORD99= 000306
T\$IBAR= 000024	WORD10= 000024	WORD42= 000124	WORD71= 000216	WRDVAL= 000310
T\$IBE= 020000	WORD11= 000026	WORD43= 000126	WORD72= 000220	XTREAD= 001000
T\$IBF= 040000	WORD12= 000030	WORD44= 000130	WORD73= 000222	XTWRT= 000400
T\$ICD= 000040	WORD13= 000032			
T\$MODE= 004000	WORD14= 000034			
T\$OB= 000036	WORD15= 000036			

. ABS. 000000 000
000000 001
CPBUG2: 002450 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3902 WORDS (16 PAGES)
DYNAMIC MEMORY: 4916 WORDS (18 PAGES)
ELAPSED TIME: 00:00:58
CPBUG2, CPBUG2/-SP=[20,1]IM,[20,1]CPBUG2

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3


```

1
2 000000      .TITLE:GREST.
3              .PSECT:CPREST.
4
5
6              HARDWARE QUERY RESOLVER, 'MANUAL' DEBUGGING AIDS.
7              CONTROL PROCESSOR TEST ROUTINES.
8
9              COMMANDS:
10             ST      STORE INTO AN CP REGISTER.
11             RE      READ FROM A REGISTER.
12             RS      RESET CP.
13             CL      CALL HDR LOADER.
14
15             ONCE A COMMAND HAS BEEN EXECUTED (OR AN ERROR ENCOUNTERED)
16             THIS MODULE RETURNS CONTROL TO THE MODULE CP AT LOCATION
17             'CPXX'.
18
19
20             STORE.
21             TABLE OF VALID REGISTER MNEMONICS AND ASSOCIATED ROUTINE
22             ADDRESSES.
23 000000
24 000000      103      122      ST2TBL:
25 000002      000400      .ASCII /CR/      ;CP CONTROL REG.
26 000004      102      113      .WORD ST2CR.
27 000006      000414      .ASCII /BK/      ;BREAKPOINT REG.
28 000010      115      101      .WORD ST2BK.
29 000012      000440      .ASCII /MA/      ;MEMORY ADDRESS REG.
30 000014      120      101      .WORD ST2MA.
31 000016      000530      .ASCII /PA/      ;PIPELINE REG. A SECT
32 000020      120      102      .WORD ST2PA.
33 000022      000562      .ASCII /PB/      ;PIPELINE REG. B SECT
34 000024      120      103      .WORD ST2PB.
35 000026      000614      .ASCII /PC/      ;PIPELINE REG. C SECT
36 000030      120      104      .WORD ST2PC.
37 000032      000646      .ASCII /PD/      ;PIPELINE REG. D SECT
38 000034      104      101      .WORD ST2PD.
39 000036      000700      .ASCII /DA/      ;DMA ADDRESS REGISTER
40 000040      104      104      .WORD ST2DA.
41 000042      000742      .ASCII /DD/      ;DMA DATA REGISTER
42 000044      122      060      .WORD ST2DD.
43 000046      001100      .ASCII /R0/      ;CP REGISTER 0
44 000050      122      061      .WORD ST2R0.
45 000052      001074      .ASCII /R1/
46 000054      122      062      .WORD ST2R1.
47 000056      001070      .ASCII /R2/
48 000060      122      063      .WORD ST2R2.
49 000062      001064      .ASCII /R3/
50 000064      122      064      .WORD ST2R3.
51 000066      001060      .ASCII /R4/
52 000070      122      065      .WORD ST2R4.
53 000072      001054      .ASCII /R5/
54 000074      122      066      .WORD ST2R5.
55 000076      001050      .ASCII /R6/
56 000100      122      067      .WORD ST2R6.
57 000102      001044      .ASCII /R7/
                    .WORD ST2R7

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

58	000104	122	070	.ASCII	/R8/	
59	000106	001040		.WORD	ST2R8	
60	000110	122	071	.ASCII	/R9/	
61	000112	001034		.WORD	ST2R9	
62	000114	122	101	.ASCII	/RA/	
63	000116	001030		.WORD	ST2RA	
64	000120	122	102	.ASCII	/RB/	
65	000122	001024		.WORD	ST2RB	
66	000124	122	103	.ASCII	/RC/	
67	000126	001020		.WORD	ST2RC	
68	000130	122	104	.ASCII	/RD/	
69	000132	001014		.WORD	ST2RD	
70	000134	122	105	.ASCII	/RE/	
71	000136	001010		.WORD	ST2RE	
72	000140	122	106	.ASCII	/RF/	
73	000142	001004		.WORD	ST2RF	
74		000031		ST2LN	==	<.-ST2TBL>4
75						
76					READ	
77					TABLE OF VALID MNEMONICS AND ASSOCIATED ROUTINE ADDRESSES	
78						
79	000144			RE2TBL:		
80	000144	115	101	.ASCII	/MA/	:READ MEMORY ADDRESS REG
81	000146	001230		.WORD	RE2MA	
82	000150	115	120	.ASCII	/MP/	:READ MICROPGM ADDRESS
83	000152	001260		.WORD	RE2MP	
84	000154	104	104	.ASCII	/DD/	:READ DMA DATA REG
85	000156	001314		.WORD	RE2DD	
86	000160	122	060	.ASCII	/R0/	:CP REGISTER 0
87	000162	001442		.WORD	RE2R0	
88	000164	122	061	.ASCII	/R1/	
89	000166	001436		.WORD	RE2R1	
90	000170	122	062	.ASCII	/R2/	
91	000172	001432		.WORD	RE2R2	
92	000174	122	063	.ASCII	/R3/	
93	000176	001426		.WORD	RE2R3	
94	000200	122	064	.ASCII	/R4/	
95	000202	001422		.WORD	RE2R4	
96	000204	122	065	.ASCII	/R5/	
97	000206	001416		.WORD	RE2R5	
98	000210	122	066	.ASCII	/R6/	
99	000212	001412		.WORD	RE2R6	
100	000214	122	067	.ASCII	/R7/	
101	000216	001406		.WORD	RE2R7	
102	000220	122	070	.ASCII	/R8/	
103	000222	001402		.WORD	RE2R8	
104	000224	122	071	.ASCII	/R9/	
105	000226	001376		.WORD	RE2R9	
106	000230	122	101	.ASCII	/RA/	
107	000232	001372		.WORD	RE2RA	
108	000234	122	102	.ASCII	/RB/	
109	000236	001366		.WORD	RE2RB	
110	000240	122	103	.ASCII	/RC/	
111	000242	001362		.WORD	RE2RC	
112	000244	122	104	.ASCII	/RD/	
113	000246	001356		.WORD	RE2RD	
114	000250	122	105	.ASCII	/RE/	

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

172 000400          ST2GR::      MOV.    BINWD,-(SP)          ;MOVE DATA TO LOD-BUS-REG.
173 000400 016746 000000G.      CALL.    CPCR              ;DIRECT CNTL WORD TO CP.
174 000404          JMP.    ST2LP.          ;TEST LOOP FLAG.
175 000410 000167 000504
176
177          :
178          :      BREAKPOINT-REG.
179 000414          ST2BK::      MOV.    #CBKCLKEN,-(SP)        ;SET CP CNTL FOR BREAKPOINT.
180 000414 012746 000400          CALL.    CPCR              ;DIRECT CNTL WORD TO CP.
181 000420          MOV.    BINWD,-(SP)        ;LOAD DATA WORD INTO LOD-BUS-REG.
182 000424 016746 000000G.      CALL.    LBOP              ;SET BCE CNTL WORD.
183 000430          JMP.    ST2LP.          ;TEST LOOP FLAG.
184 000434 000167 000460
185
186          :
187          :      MEMORY ADDRESS-REG.
188 000440          ST2MA::      MOV.    #<PLRWR+PLD>,-(SP)      ;DIRECT CNTL WORD TO CP.
189 000440 012746 000230          CALL.    CPCR              ;SEND MAR LOD-BIT.
190 000444          MOV.    #<MARLOD>,-(SP)      ;SEND DATA TO CP.
191 000450 012746 010000          CALL.    LBOP              ;SET MAREN-BIT AND ALSO.
192 000454          MOV.    #<MREN1>,-(SP)      ;CLEAR PLR-D-ENABLE BITS.
193 000460 012746 000001          CALL.    CPCR              ;MOVE DATA WORD TO LOD-BUS-REG.
194 000464          MOV.    BINWD,-(SP)      ;SEND DATA TO CP (MAR)
195 000470 016746 000000G.      CALL.    LBOP              ;DIRECT CNTL WORD TO CP-CR.
196 000474          MOV.    #<PLRWR+PLD>,-(SP)      ;CLEAR PLR-D-BITS.
197 000500 012746 000230          CALL.    CPCR              ;SEND DATA TO CP.
198 000504          CLR.    -(SP)              ;CLEAR CP-CR-BITS.
199 000510 005046          CALL.    LBOP              ;DIRECT CNTL WORD TO CP-CR.
200 000512          CLR.    -(SP)              ;CLEAR PLR-D-BITS.
201 000516 005046          CALL.    LBOP              ;SEND DATA TO CP.
202 000520          CLR.    -(SP)              ;CLEAR CP-CR-BITS.
203 000524 000167 000370          CALL.    CPCR              ;DIRECT CNTL WORD TO CP-CR.
204          JMP.    ST2LP.
205
206          :
207          :      CP PIPELINE-REG-SECTION-A.
208          :
209          :      NOTE:  IF CSR-#2 IS SET-UP FOR SEARCH MODE FOR ANY REASON, THESE
210          :      4 PIPELINE-ROUTINES WILL NOT WORK!!!!
211 000530          ST2PA::      MOV.    #PLRWR,-(SP)
212 000530 012746 000200          CALL.    CPCR              ;DIRECT CNTL WORD TO CP.
213 000534          MOV.    BINWD,-(SP)        ;MOVE DATA WORD TO LOD-BUS-REG.
214 000540 016746 000000G.      CALL.    LBOP              ;SEND DATA TO CP.
215 000544          CLR.    -(SP)
216 000550 005046          CALL.    CPCR              ;CLEAR THE CONTROL REGISTER.
217 000552 000167 000336          JMP.    ST2LP.          ;TEST FOR REPEAT.
218
219          :
220          :      CP PIPELINE-REG-SECTION-B.
221          :      CAUTION: DO NOT SET-BIT-14 DURING THIS TEST. IT WILL LOCK-
222          :      UP THE BUS.
223          :      ALSO DO NOT USE XXX1, XXX2, XXX6, XXX7, OR XXXA.
224 000562          ST2PB::      MOV.    #<PLRWR+PLB>,-(SP)
225 000562 012746 000210          CALL.    CPCR              ;DIRECT CNTL WORD TO CP.
226 000566          MOV.    BINWD,-(SP)        ;MOVE DATA WORD TO LOD-BUS-REG.
227 000572 016746 000000G.      CALL.    LBOP              ;SEND DATA TO CP.
228 000576          CLR.    -(SP)
229 000582 005046          CALL.    CPCR

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
229 000604          CALL    CPCR          ; CLEAR THE CONTROL REGISTER  NS
230 000610 000167 000304      JMP     ST2LP          ; TEST FOR REPEAT
231
232
233          CP PIPELINE REG SECTION C
234          CAUTION: DO NOT SET BIT 14 DURING THIS TEST. IT WILL LOCK
235          UP THE BUS.
236 000614          ST2PC::
237 000614 012746 000220      MOV     #<PLRWR+PLC>,-(SP)
238 000620          CALL    CPCR          ; DIRECT CNTL WORD TO CP
239 000624 016746 000000G      MOV     BINWD,-(SP)      ; MOVE DATA WORD TO LOD BUS REG
240 000630          CALL    LBCP          ; SEND DATA TO CP
241 000634 005046          CLR     -(SP)
242 000636          CALL    CPCR          ; CLEAR THE CONTROL REGISTER  NS
243 000642 000167 000252      JMP     ST2LP          ; TEST FOR REPEAT
244
245          CP PIPELINE REG SECTION D
246          CAUTION: DO NOT SET BIT 11 DURING THIS TEST. IT WILL LOCK
247          UP THE BUS.
248
249 000646          ST2PD::
250 000646 012746 000230      MOV     #<PLRWR+PLD>,-(SP)
251 000652          CALL    CPCR          ; DIRECT CNTL WORD TO CP
252 000656 016746 000000G      MOV     BINWD,-(SP)      ; MOVE DATA WORD TO LOD BUS REG
253 000662          CALL    LBCP          ; SEND DATA TO CP
254 000666 005046          CLR     -(SP)
255 000670          CALL    CPCR          ; CLEAR THE CONTROL REGISTER  NS
256 000674 000167 000220      JMP     ST2LP          ; TEST FOR REPEAT
257
258          WRITE DMA ADDRESS REGISTER
259
260 000700          ST2DA::
261 000700 012746 000210      MOV     #<PLRWR+PLB>,-(SP)
262 000704          CALL    CPCR          ; SET UP PLR B
263 000710 012746 000005      MOV     #DMAWRH,-(SP)      ; SET VALUE FOR CONTROL WORD
264 000714          CALL    LBCP          ; SEND WORD TO CP
265 000720 016746 000000G      MOV     BINWD,-(SP)      ; SET VALUE FOR DMA REG
266 000724          CALL    LBCP          ; TRANSFER TO CP
267 000730 005046          CLR     -(SP)      ; CLEAR CP CONTROL REG
268 000732          CALL    CPCR
269 000736 000167 000156      JMP     ST2LP
270
271          WRITE DMA DATA REGISTER
272
273 000742          ST2DD::
274 000742 012746 000210      MOV     #<PLRWR+PLB>,-(SP)
275 000746          CALL    CPCR          ; SET UP PLR B
276 000752 012746 000004      MOV     #DMAWRH,-(SP)      ; SET VALUE FOR CONTROL WORD
277 000756          CALL    LBCP          ; SEND WORD TO CP
278 000762 016746 000000G      MOV     BINWD,-(SP)      ; SET VALUE FOR DMA REG
279 000766          CALL    LBCP          ; TRANSFER TO CP
280 000772 005046          CLR     -(SP)      ; CLEAR CP CONTROL REG
281 000774          CALL    CPCR
282 001000 000167 000114      JMP     ST2LP
283
284          LOAD CP REGISTERS
285
```

```

286 001004 005267 000000G ST2RF:: INC INDEX
287 001010 005267 000000G ST2RE:: INC INDEX
288 001014 005267 000000G ST2RD:: INC INDEX
289 001020 005267 000000G ST2RC:: INC INDEX
290 001024 005267 000000G ST2RB:: INC INDEX
291 001030 005267 000000G ST2RA:: INC INDEX
292 001034 005267 000000G ST2R9:: INC INDEX
293 001040 005267 000000G ST2R8:: INC INDEX
294 001044 005267 000000G ST2R7:: INC INDEX
295 001050 005267 000000G ST2R6:: INC INDEX
296 001054 005267 000000G ST2R5:: INC INDEX
297 001060 005267 000000G ST2R4:: INC INDEX
298 001064 005267 000000G ST2R3:: INC INDEX
299 001070 005267 000000G ST2R2:: INC INDEX
300 001074 005267 000000G ST2R1:: INC INDEX
301 001100 ST2R0:: INC INDEX
302 ;
303 001100 016700 000000G MO:: INDEX,R0 ;PREPARE TO CALL SUBRTN
304 001104 CALL LDREG ;LOAD REGISTER
305 001110 005067 000000G CLR INDEX
306 001114 000167 000000 JMP ST2LP
307 ;
308 ;
309 ;
310 001120 ST2LP:
311 001120 032767 000000G 000000G BIT #LOOP,BASE ;REPEAT ?
312 001126 001402 BEQ ST2X ;NO
313 001130 000167 177234 JMP ST2IN
314 ;
315 001134 ST2X:
316 001134 CALL KILL ;KILL AST
317 001140 000167 000000G JMP CPXX

```

```

319      ;
320      ;
321      ;      READ.
322      ;      PERFORM THIRD-LEVEL PARSING.
323      ;      EG. IN THE COMMAND:
324      ;      CP>RE CR.
325      ;      PARSE THE 'CR'
326      ;
327      ;
328      001144      RE2::
329      001144      CALL.   FIND           ;FIND A REG. MNEMONIC IN COMMAND LINE.
330      001150      BCC.    1$           ;OK, CONTINUE
331      001152      CALL.   ERR4          ;NOTHING THERE.
332      001156      000167 000346      JMP.    RE2X          ;RETURN TO TOP OF LOOP (PROMPT)
333      ;
334      ;
335      ;      MATCH THE REGISTER MNEMONIC FROM THE COMMAND LINE AGAINST
336      ;      THE TABLE OF VALID MNEMONICS.
337      001162      012700 000023      1$:   MOV.    #RE2LN,R0          ;NUMBER OF TABLE ENTRIES.
338      001166      012702 000144      MOV.    #RE2TBL,R2          ;R2 -> TABLE.
339      001172      CALL.   SCAN          ;MATCH AGAINST COMMAND LINE.
340      001176      103004      BCC.    2$           ;OK, CONTINUE.
341      001200      CALL.   ERR6          ;
342      001204      000167 000320      JMP.    RE2X          ;
343      ;
344      ;
345      ;      SAVE THE POINTER TO THE ROUTINE ASSOCIATED WITH THE
346      ;      REGISTER. R1 -> ROUTINE ADDRESS.
347      ;      CALL ROUTINE TO SCAN COMMAND LINE FOR LOOP INDICATOR.
348      ;      EG. CP>RE CR L.
349      ;      LOOP FLAG WILL BE SET IF INDICATOR IS PRESENT.
350      ;      JUMP TO ROUTINE TO LOAD REGISTER.
351      001210      010167 000000G      2$:   MOV.    R1,RTNPT          ;SAVE POINTER TO RTN
352      001214      CALL.   LOOPR          ;LOOP?
353      001220      016701 000000G      RE2IN: MOV.    RTNPT,R1          ;POINT TO ROUTINE.
354      001224      000171 000000      JMP.    @R1           ;EXECUTE ROUTINE.
355      ;
356      ;
357      ;      MEMORY ADDRESS REG.
358      ;
359      001230      RE2MA::
360      001230      012746 000002      MOV.    #<MAROUT>,-(SP)
361      001234      CALL.   CPCR          ;DIRECT CNTL WORD TO CP.
362      001240      CALL.   CPLB          ;REQUEST CP TO LOD BUS.
363      001244      005046      CLR.    -(SP)          ;CLEAR THE CONTROL REGISTER. NS.
364      001246      CALL.   CPCR          ;
365      001252      012601      MOV.    (SP)+,R1          ;CP WORD RETURNED ON STACK. NS.
366      001254      000167 000202      JMP.    RE2PUT          ;
367      ;
368      ;
369      ;      MICROPGM ADDRESS REG.
370      001260      RE2MP::
371      001260      012746 000004      MOV.    #CSADR,-(SP)
372      001264      CALL.   CPCR          ;DIRECT CNTL WORD TO CP.
373      001270      CALL.   CPLB          ;REQUEST CP TO LOD BUS.
374      001274      005046      CLR.    -(SP)          ;CLEAR THE CONTROL REGISTER. NS.
375      001276      CALL.   CPCR          ;

```

```

376 001302 012601      MOV.    (SP)+,R1      ;CP WORD RETURNED ON STACK
377 001304 042701 176000 BIC.    *176000,R1      ;CLEAR USELESS BITS
378 001310 000167 000146 JMP.    RE2PUT
379
380
381      ; READ DMA DATA REGISTER
382 001314      RE2DD::
383 001314 012746 000210 MOV.    *(<PLWR+PLB>,-(SP)
384 001320      CALL.    CPCR      ;DIRECT CNTL WORD TO CP
385 001324 012746 000003 MOV.    *DMARRDH,-(SP)    ;SET VALUE FOR CONTROL WORD
386 001330      CALL.    LBCP      ;SEND DATA TO CP
387 001334      CALL.    CPLB      ;READ FROM CP
388 001340 012601      MOV.    (SP)+,R1      ;LOAD VALUE INTO R1 FOR PRINT
389 001342 000167 000114 JMP.    RE2PUT
390
391      ; PRINT CP REGISTERS
392
393 001346 005267 000000G RE2RF:: INC.    INDEX
394 001352 005267 000000G RE2RE:: INC.    INDEX
395 001356 005267 000000G RE2RD:: INC.    INDEX
396 001362 005267 000000G RE2RC:: INC.    INDEX
397 001366 005267 000000G RE2RB:: INC.    INDEX
398 001372 005267 000000G RE2RA:: INC.    INDEX
399 001376 005267 000000G RE2R9:: INC.    INDEX
400 001402 005267 000000G RE2R8:: INC.    INDEX
401 001406 005267 000000G RE2R7:: INC.    INDEX
402 001412 005267 000000G RE2R6:: INC.    INDEX
403 001416 005267 000000G RE2R5:: INC.    INDEX
404 001422 005267 000000G RE2R4:: INC.    INDEX
405 001426 005267 000000G RE2R3:: INC.    INDEX
406 001432 005267 000000G RE2R2:: INC.    INDEX
407 001436 005267 000000G RE2R1:: INC.    INDEX
408 001442      RE2R0::
409
410 001442 016700 000000G MOV.    INDEX,R0      ;PREPARE TO CALL SUBRTN
411 001446      CALL.    REREG      ;READ A REGISTER
412 001452 005067 000000G CLR.    INDEX
413 001456 000167 000000G JMP.    RE2PUT
414
415
416 001462 032767 000000G-000000G RE2PUT: BIT.    #ONCE,BASE
417 001470 001011      BNE.    1$      ;PRINTED ONCE ?
418 001472 052767 000000G-000000G BIS.    #ONCE,BASE
419 001500 012705 000000G MOV.    #PRINT,R5      ;YES, SKIP
420 001504      CALL.    UNPK      ;SET FLAG FOR PRINTED ONCE
421 001510      CALL.    CONSOL      ;POINT TO PRINT LINE
422
423 001514 032767 000000G-000000G 1$: BIT.    #LOOP,BASE
424 001522 001402      BEQ.    RE2X      ;REPEAT
425 001524 000167 177470 JMP.    RE2IN      ;NO, EXIT
426
427 001530      RE2X:
428 001530 042767 000000G-000000G BIC.    #ONCE,BASE      ;CLEAR PRINT CONTROL FLAG
429 001536      CALL.    KILL      ;KILL
430 001542 000167 000000G JMP.    CPXX

```



```
432.      :  
433.      :  
434.      :      RESET CP  
435.      :  
436.      :  
437 001546      RS2::  
438 001546 005046      CLR      -(SP)      ;RESET NOTHING  
439 001550 012746 000002      MOV      #0$CSET, -(SP)      ;SET CP RESET  
440 001554      CALL      CSRI      ;SEND  
441 001560 012746 000002      MOV      #0$CSET, -(SP)      ;CLEAR CP RESET  
442 001564 005046      CLR      -(SP)      ;SET NOTHING  
443 001566      CALL      CSRI      ;DO IT  
444 001572 000167 000000G      JMP      CPXX      FW  
445      :  
446      :  
447      :  
448      :      CALL HQR LOADER  
449      :  
450      :  
451 001576      CL2::  
452 001576      CALL      CL      ;CALL ROUTINE IN MAIN  
453 001602 000167 000000G      JMP      CPXX  
454      :  
455      000001      .END
```

ALUCKE = 040000	BYTE40 = 000050	BYTE92 = 000134	MMLEFT = 000002	Q\$MSET = 000004
ALUOE = 004000	BYTE41 = 000051	BYTE93 = 000135	MMOE = 000004	Q\$MSP = 100000
A01 = 010000	BYTE42 = 000052	BYTE94 = 000136	MMWRTE = 000010	Q\$NCLK = 176000
BASE = 000000	BYTE43 = 000053	BYTE95 = 000137	MNOBRE = 100000	Q\$PP = 000100
BINWD = 000000	BYTE44 = 000054	BYTE96 = 000140	MREN1 = 000001	Q\$PPSW = 000320
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	MREN2 = 020000	Q\$PP2 = 000300
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	MSYN = 000040	Q\$QHLT = 000013
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	N = 000144	Q\$QL = 000043
BIT10 = 000200	BYTE48 = 000060	BYTVAL = 000144	ONCE = 000000	Q\$QLA = 000053
BIT11 = 000400	BYTE49 = 000061	CBKALL = 001000	PACK = 000000	Q\$QLB = 000054
BIT12 = 010000	BYTE5 = 000005	CBKCLK = 000400	PLB = 000010	Q\$QLR = 000001
BIT13 = 020000	BYTE50 = 000062	CL = 001576RG	PLC = 000020	Q\$QW = 000042
BIT14 = 040000	BYTE51 = 000063	CL2 = 001576RG	PLD = 000030	Q\$RDCD = 000005
BIT15 = 100000	BYTE52 = 000064	CNOBRE = 100000	PLRWR = 000200	Q\$RDMD = 000006
BIT2 = 000004	BYTE53 = 000065	CONSOL = 000000	PLR.EN = 000200	Q\$REBK = 001000
BIT3 = 000010	BYTE54 = 000066	CPCCEN = 010000	PRINT = 000000	Q\$RNC = 006000
BIT4 = 000020	BYTE55 = 000067	CPCR = 000000	Q\$RSC = 004000	Q\$RSET = 000010
BIT5 = 000040	BYTE56 = 000070	CPLB = 000000	Q\$RSM = 100000	Q\$SP = 000120
BIT6 = 000100	BYTE57 = 000071	CPREAD = 040000	Q\$SP2 = 000340	REREG = 000000
BIT7 = 000200	BYTE58 = 000072	CPWRTE = 020000	RE2 = 001144RG	RE2DD = 001314RG
BIT8 = 000400	BYTE59 = 000073	CPXX = 000000	RE2IN = 001220RG	RE2LN = 000023 G
BIT9 = 001000	BYTE6 = 000006	CSADRD = 000004	RE2MA = 001230RG	RE2MP = 001260RG
BYTE0 = 000000	BYTE60 = 000074	CSEQCI = 100000	RE2MPUT = 001462RG	RE2RA = 001372RG
BYTE1 = 000001	BYTE61 = 000075	CSOE = 000040	RE2RB = 001366RG	RE2RC = 001362RG
BYTE10 = 000012	BYTE62 = 000076	CSR1 = 000000	RE2RD = 001356RG	RE2RE = 001352RG
BYTE11 = 000013	BYTE63 = 000077	CSWRTE = 000100	RE2RF = 001346RG	RE2R0 = 001442RG
BYTE12 = 000014	BYTE64 = 000100	DBR.RD = 000001	RE2R1 = 001436RG	RE2R2 = 001432RG
BYTE13 = 000015	BYTE65 = 000101	DB\$CPP = 001457	RE2R3 = 001426RG	RE2R4 = 001422RG
BYTE14 = 000016	BYTE66 = 000102	DB\$SPT = 000026	RE2R5 = 001416RG	RE2R6 = 001412RG
BYTE15 = 000017	BYTE67 = 000103	DE\$TPC = 000023	RE2R7 = 001406RG	RE2R8 = 001402RG
BYTE16 = 000020	BYTE68 = 000104	DISPGS = 100000	RE2R9 = 001376RG	RE2TBL = 000144RG
BYTE17 = 000021	BYTE69 = 000105	DMAWR = 000005	RE2X = 001530RG	RGQ.EN = 000200
BYTE18 = 000022	BYTE7 = 000007	DMARRD = 000003	RGQ.VA = 020000	RS2 = 001546RG
BYTE19 = 000023	BYTE70 = 000106	DMARRW = 000004	RTNPT = 000000	SCAN = 000000
BYTE2 = 000002	BYTE71 = 000107	ENBR = 010000	SEQ.CI = 000010	ST2BK = 000414RG
BYTE20 = 000024	BYTE72 = 000110	ERR4 = 000000	ST2CR = 000400RG	ST2DA = 000700RG
BYTE21 = 000025	BYTE73 = 000111	ERR5 = 000000		
BYTE22 = 000026	BYTE74 = 000112	ERR6 = 000000		
BYTE23 = 000027	BYTE75 = 000113	FIND = 000000		
BYTE24 = 000030	BYTE76 = 000114	INDEX = 000000		
BYTE25 = 000031	BYTE77 = 000115	KILL = 000000		
BYTE26 = 000032	BYTE78 = 000116	LBCP = 000000		
BYTE27 = 000033	BYTE79 = 000117	LDREG = 000000		
BYTE28 = 000034	BYTE8 = 000010	LOC.EN = 000100		
BYTE29 = 000035	BYTE80 = 000120	LOC.WA = 040000		
BYTE3 = 000003	BYTE81 = 000121	LOC.WB = 100000		
BYTE30 = 000036	BYTE82 = 000122	LOOP = 000000		
BYTE31 = 000037	BYTE83 = 000123	LOOPR = 000000		
BYTE32 = 000040	BYTE84 = 000124	MAREN1 = 000001		
BYTE33 = 000041	BYTE85 = 000125	MAREN2 = 004000		
BYTE34 = 000042	BYTE86 = 000126	MARLOD = 010000		
BYTE35 = 000043	BYTE87 = 000127	MAROUT = 000002		
BYTE36 = 000044	BYTE88 = 000130	MAR.LO = 002000		
BYTE37 = 000045	BYTE89 = 000131	MAR.OU = 000040		
BYTE38 = 000046	BYTE9 = 000011	MBKALL = 001000		
BYTE39 = 000047	BYTE90 = 000132	MBKCLK = 000400		
BYTE4 = 000004	BYTE91 = 000133	MMADRD = 000100		

CPREST: M1110 27-MAR-80 14:44 PAGE:7-2
SYMBOL: T

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

ST2DD: 000742RG	002: TD\$QAR= 176344	T\$SC: = 000022	WORD33= 000102	WORD68= 000210
ST2IN: 000370R	002: TD\$QTR= 176346	T\$SCLK= 020000	WORD34= 000104	WORD69= 000212
ST2LN: 000031 G	TD\$QRD= 000274	T\$SEG1= 000000	WORD35= 000106	WORD7= 000016
ST2LP: 001120R	002: TD\$SW= 176376	T\$SEG2= 000001	WORD36= 000110	WORD70= 000214
ST2MA: 000440RG	002: TD\$TAR= 176372	T\$SEG3= 000002	WORD37= 000112	WORD71= 000216
ST2PA: 000530RG	002: TD\$TAW= 176362	T\$SO: = 000001	WORD38= 000114	WORD72= 000220
ST2PB: 000562RG	002: TD\$TDR= 176374	T\$UBUS= 100000	WORD39= 000116	WORD73= 000222
ST2PC: 000614RG	002: TD\$TDW= 176364	T\$1CLK= 000400	WORD4= 000010	WORD74= 000224
ST2PD: 000646RG	002: T\$AD= 000020	T\$BEN= 000020	WORD40= 000120	WORD75= 000226
ST2RA: 001030RG	002: T\$BA= 000002	UBD: IN= 000020	WORD41= 000122	WORD76= 000230
ST2RB: 001024RG	002: T\$BD= 000010	UNPK: = ***** GX	WORD42= 000124	WORD77= 000232
ST2RC: 001020RG	002: T\$BSO= 100000	WORD0= 000000	WORD43= 000126	WORD78= 000234
ST2RD: 001014RG	002: T\$BT= 000020	WORD1= 000002	WORD44= 000130	WORD79= 000236
ST2RE: 001010RG	002: T\$BTAR= 000030	WORD10= 000024	WORD45= 000132	WORD8= 000020
ST2RF: 001004RG	002: T\$BTD= 002000	WORD11= 000026	WORD46= 000134	WORD80= 000240
ST2R0: 001100RG	002: T\$CD= 000100	WORD12= 000030	WORD47= 000136	WORD81= 000242
ST2R1: 001074RG	002: T\$CLK= 002000	WORD13= 000032	WORD48= 000140	WORD82= 000244
ST2R2: 001070RG	002: T\$DISK= 000200	WORD14= 000034	WORD49= 000142	WORD83= 000246
ST2R3: 001064RG	002: T\$DRD= 000004	WORD15= 000036	WORD5= 000012	WORD84= 000250
ST2R4: 001060RG	002: T\$EMEM= 010000	WORD16= 000040	WORD50= 000144	WORD85= 000252
ST2R5: 001054RG	002: T\$FSA= 000000	WORD17= 000042	WORD51= 000146	WORD86= 000254
ST2R6: 001050RG	002: T\$FSAB= 000004	WORD18= 000044	WORD52= 000150	WORD87= 000256
ST2R7: 001044RG	002: T\$FSAC= 000014	WORD19= 000046	WORD53= 000152	WORD88= 000260
ST2R8: 001040RG	002: T\$FSB2= 000010	WORD2= 000004	WORD54= 000154	WORD89= 000262
ST2R9: 001034RG	002: T\$IB= 000026	WORD20= 000050	WORD55= 000156	WORD9= 000022
ST2TBL: 000000RG	002: T\$IBAR= 000024	WORD21= 000052	WORD56= 000160	WORD90= 000264
ST2X: 001134R	002: T\$IBE= 020000	WORD22= 000054	WORD57= 000162	WORD91= 000266
S\$CLR= 000000	T\$IBF= 040000	WORD23= 000056	WORD58= 000164	WORD92= 000270
S\$LA= 000001	T\$ICD= 000040	WORD24= 000060	WORD59= 000166	WORD93= 000272
S\$QB= 000005	T\$MODE= 004000	WORD25= 000062	WORD6= 000014	WORD94= 000274
S\$QR= 000006	T\$OB= 000036	WORD26= 000064	WORD60= 000170	WORD95= 000276
S\$QX= 000004	T\$OBE= 004000	WORD27= 000066	WORD61= 000172	WORD96= 000300
S\$SR= 000007	T\$OBF= 010000	WORD28= 000070	WORD62= 000174	WORD97= 000302
S\$S1= 000010	T\$OBRA= 000034	WORD29= 000072	WORD63= 000176	WORD98= 000304
S\$S2= 000014	T\$OBWA= 000032	WORD3= 000006	WORD64= 000200	WORD99= 000306
TD\$CTR= 176370	T\$OUTH= 100000	WORD30= 000074	WORD65= 000202	WORDVAL= 000310
TD\$CTW= 176360	T\$RBD= 000200	WORD31= 000076	WORD66= 000204	XTREAD= 001000
TD\$INL= 004000	T\$RNB= 000040	WORD32= 000100	WORD67= 000206	XTURTE= 000400
TD\$MEM= 000270	T\$RSET= 040000			

. ABS: 000000 000
000000 001
CPREST: 001606 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3426 WORDS (14 PAGES)
DYNAMIC MEMORY: 4916 WORDS (18 PAGES)
ELAPSED TIME: 00:00:50
CPREST,CPREST--SP=C20.1JIM,C20.1JCPREST

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

1      .TITLE=GRLD-
2 000000 .PSECT=CPLD-
3
4
5      ;
6      ;
7      ;
8      ;
9      ;
10     ;
11     ;
12     ;
13     ;
14     ;
15     ;
16     ;
17     ;
18     ;
19     ;
20     ;
21     ;
22     ;
23     ;
24     ;
25     ;
26     ;
27     ;
28     ;
29     ;
30     ;
31     ;
32     ;
33     ;
34     ;
35     ;
36     ;
37     ;
38 000000 LD2TBL::
39 000000      103      123      .ASCII /CS/      ;LOAD MICROPGM MEMORY
40 000002      000272'      .WORD LD2CS      ;
41 000004      103      104      .ASCII /CD/      ;LOAD DATA MEMORY
42 000006      000472'      .WORD LD2CD      ;
43 000010      110      114      .ASCII /HL/      ;LOAD HRL BUFFER
44 000012      000734'      .WORD LD2HL      ;
45 000014      102      114      .ASCII /BL/      ;LOAD BCL
46 000016      001114'      .WORD LD2BL      ;
47 000004      LD2LN      ==      <.-LD2TBL>/4
48
49     ;
50     ;
51     ;
52     ;
53     ;
54     ;
55     ;
56     ;
57 000020

```

HARDWARE QUERY RESOLVER "MANUAL" DEBUGGING AIDS
 CONTROL PROCESSOR TEST ROUTINES
 ***** PROTOTYPE VERSION *****
 MEMORY LOAD ROUTINES

COMMANDS:
 LD LOAD CP MEMORIES

ONCE A COMMAND HAS BEEN EXECUTED (OR AN ERROR ENCOUNTERED)
 THIS MODULE RETURNS CONTROL TO THE MODULE MRP AT LOCATION
 'CPXX'

NOTE:
 THE COMMAND TO LOAD CP DATA MEMORY CANNOT BE USED
 UNLESS MRP MICROPROGRAM MEMORY AND CP CONTROL STORE
 HAVE BEEN LOADED VIA THE "LOADER" PROGRAM

IF THE QEX MEMORIES ARE EITHER LOADED OR PRINTED AFTER
 CP DATA MEMORY HAS BEEN LOADED, CP DATA MEMORY CONTENTS
 ARE LOST

.MCALL WTSE\$S,CLEF\$S

TABLE OF VALID CP MEMORY MNEMONICS AND ADDRESSES OF
 ASSOCIATED ROUTINES

LD2TBL::

Address	Value	Label	Description
000000	103	123	LOAD MICROPGM MEMORY
000002	000272'		
000004	103	104	LOAD DATA MEMORY
000006	000472'		
000010	110	114	LOAD HRL BUFFER
000012	000734'		
000014	102	114	LOAD BCL
000016	001114'		
000004			

LD2LN == <.-LD2TBL>/4

LOAD
 PERFORM THIRD LEVEL PARSING
 EG. IN THE COMMAND:
 CP>LD CD 0
 PARSE TH "CD"

LD2::

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

58 000020          CALL    FIND                ;LOCATE THE MEMORY MNEMONIC IN THE COMMAND LINE.
59 000024 103004    BCC     1$                 ;OK, CONTINUE.
60 000026          CALL    ERR4                ;MISSING OPERAND.
61 000032 000167 001236 JMP     LD2X             ;EXIT.
62          ;
63          ;
64          ;
65 000036 012700 000004 1$: MOV     #LD2LN,R0          ;NUMBER OF TABLE ENTRIES.
66 000042 012702 000000 MOV     #LD2TBL,R2        ;POINT TO TABLE.
67 000046          CALL    SCAN                ;MATCH AGAINST COMMAND LINE.
68 000052 103004    BCC     2$                 ;MATCH WAS MADE.
69 000054          CALL    ERR7                ;INVALID MEMORY MNEMONIC.
70 000060 000167 001210 JMP     LD2X
71          ;
72          ;
73          ;
74          ;
75          ;
76          ;
77          ;
78          ;
79          ;
80          ;
81          ;
82          ;
83          ;
84          ;
85          ;
86          ;
87          ;
88          ;
89          ;
90          ;
91          ;
92          ;
93 000064 010167 000000G 2$: MOV     R1,RTNPT        ;SAVE POINTER.
94 000070          CALL    FIND                ;LOCATE START ADDRESS IN COMMAND LINE.
95 000074 103004    BCC     3$                 ;OK, CONTINUE.
96 000076          CALL    ERR4                ;MISSING OPERAND.
97 000102 000167 001166 JMP     LD2X             ;EXIT.
98 000106          CALL    PACK                ;CONVERT COMMAND LINE VALUE TO BINARY.
99 000112 103004    BCC     4$                 ;CONVERSION SUCCESSFUL.
100 000114          CALL    ERR5                ;INVALID NUMERIC VALUE.
101 000120 000167 001150 JMP     LD2X
102          ;
103 000124 016767 000000G 000000G 4$: MOV     BINWD,MSTR1        ;SAVE LOADING START ADDRESS.
104 000132 016767 000000G 000000G MOV     BINWD,MSTR2        ;SAVE IT TWICE (FOR REFRESH ON LOOP)
105 000140 012767 177777 000000G MOV     #-1,MEND        ;INIT END ADDRESS.
106          ;
107          ;
108          ;
109          ;
110          ;
111 000146          CALL    FIND                ;SCAN COMMAND LINE.
112 000152 103004    BCC     5$                 ;SOMETHING THERE.
113 000154 052767 000000G 000000G BIS     #RP,BASE        ;SIGNAL TO REPEAT PROMPT.
114 000162 000437    BR      9$                 ;JUMP TO RTN.

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

115 ;
116 000164 122711 000114 5$: CMPB #1L,(R1) ;LOOP INDICATOR
117 000170 001006 BNE 6$ ;NO. MUST BE UPPER ADDRESS
118 000172 016767 000000G-000000G MOV MSTRT,MEND ;SET END ADDR = START ADDR
119 000200 CALL HANG ;HOW TO STOP LOOP
120 000204 000426 BR 9$ ;JUMP TO RTN
121 ;
122 000206 6$: CALL PACK ;CONVERT UPPER ADDRESS
123 000212 103004 BCC 7$ ;OK, CONTINUE
124 000214 CALL ERR5 ;INVALID NUMERIC
125 000220 000167 001050 JMP LD2X ;EXIT
126 ;
127 ; SAVE END ADDRESS (BINARY)
128 ; CHECK FOR LOOP INDICATOR AFTER END ADDRESS (CONDITION 4)
129 ;
130 000224 016767 000000G-000000G 7$: MOV BINWD,MEND ;SET UP END ADDRESS
131 000232 CALL FIND ;CHECK FOR LOOP INDICATOR
132 000236 103411 BCS 9$ ;NO LOOP
133 000240 122711 000114 CMPB #1L,(R1) ;CORRECT INDICATOR
134 000244 001404 BEQ 8$ ;YES, CONTINUE
135 000246 CALL ERR11 ;LOOP OPTION ERROR
136 000252 000167 001016 JMP LD2X
137 000256 8$: CALL HANG ;HOW TO STOP LOOP
138 ;
139 000262 016701 000000G 9$: MOV RTNPT,R1 ;POINT TO ROUTINE
140 000266 000171 000000 JMP @R1 ;JUMP TO ROUTINE
141 ;
142 ;
143 ; LOAD CONTROL STORE
144 ;
145 ;
146 000272 LD2GS:
147 000272 005046 CLR -(SP) ;CLEAR NOTHING
148 000274 012746 000002 MOV #0$CSET,-(SP) ;CP RESET
149 000300 CALL CSR1 ;DO IT
150 000304 012746 000002 MOV #0$CSET,-(SP) ;CLEAR CP RESET
151 000310 005046 CLR -(SP) ;SET NOTHING
152 000312 CALL CSR1
153 ;
154 000316 016746 000000G MOV CSHIGH,-(SP) ;SUPPLY UPPER MEMORY LIMIT
155 000322 016746 000000G MOV CSLOW,-(SP) ;LOWER LIMIT
156 000326 CALL BUF54 ;PREPARE TO LOAD
157 000332 103002 BCC 100$ ;NO ERRORS, CONTINUE
158 000334 000167 000734 JMP LD2X
159 ;
160 000340 032767 000000G-000000G 100$: BIT #RP,BASE ;REPEAT PROMPT
161 000346 001416 BEQ 2$ ;NO, ONCE ONLY
162 000350 1$: CALL PDATA ;READ DATA FROM COMMAND LINE
163 000354 103002 BCC 10$ ;NOT END OF MEMORY
164 000356 000167 000712 JMP LD2X ;IF END, EXIT
165 000362 102002 BVC 20$ ;<CR> RESPONSE TO PROMPT
166 000364 000167 000704 JMP LD2X ;EXIT ON <CR> RESPONSE
167 000370 20$: CALL LOADCS ;LOAD 4 SECTIONS
168 000374 066767 000000G-000000G ADD INCVAL,MSTR2 ;BUMP LOAD ADDRESS
169 000402 000762 BR 1$ ;REPEAT
170 ;
171 ; PROMPT ONCE, THEN FILL MEMORY

```

```
172.
173 000404          2$: CALL PDATA
174 000410 103002.   BCC 30$      ;PROMPT
175 000412 000167   JMP LD2X      ;NOT-END-OF-MEMORY
176 000416 102002.   BVC 3$       ;ELSE-EXIT
177 000420 000167   JMP LD2X      ;NOT- $\langle$ CR $\rangle$  RESPONSE
178
179 000424          3$: CALL LOADCS
180 000430 066767 000000G-000000G. ADD INCVAL,MSTR2
181 000436 026767 000000G-000000G. CMP MSTR2,MEND
182 000444 101767   BLOS 3$       ;LOAD-4 SECTIONS
183 000446 032767 000000G-000000G. BIT *LOOP,BASE ;ADVANCE ADDRESS
184 000454 001002.   BNE 4$       ;HAS UPPER MEMORY LIMIT BEEN REACHED
185 000456 000167 000612   JMP LD2X ;NO CONTINUE
186 000462 016767 000000G-000000G.4$: MOV MSTR2,MSTR2 ;LOOP ON?
187 000470 000755   BR 3$         ;YES CONTINUE
188                                     ;NO LOOP EXIT
189                                     ;REINITIALIZE ADDRESS
190
191          LOAD DATA MEMORY
192 000472.          LD2CD:
193 000476 016746 000000G. MOV CDHIGH,-(SP)
194 000476 016746 000000G. MOV CDLOW,-(SP)
195 000502.          CALL BUFSET ;SUPPLY UPPER MEMORY LIMIT
196 000506 103002.   BCC 100$      ;LOWER LIMIT
197 000510 000167 000560   JMP LD2X ;PREPARE FOR LOAD
198                                     ;NO ERROR, CONTINUE
199 000514 005046   100$: CLR -(SP) ;START MICROCODE AT 0
200 000516          CALL SEQCS
201
202 000522 005046   CLR -(SP)
203 000524          CALL CPCR ;REINHIBIT BRANCH CONTROL REGISTER
204
205 000530 012746 000377   MOV #377,-(SP)
206 000534          CALL SEQMM ;SET MRP MICRO ADDRESS = X'FF' (JUMP SELF)
207
208 000540 005046   CLR -(SP)
209 000542          CALL MRPCR ;REINHIBIT BRANCH CONTROL REGISTER
210
211 000546 012767 001000 176422. MOV #Q$REBK,Q$CR2
212 000554 012767 120000 176422. MOV *Q$SM+Q$ENOP,Q$CR2
213 000562 012746 000360   MOV #Q$CSEL,-(SP) ;RE-ARM INTERRUPTS
214 000566 052716 001001   BIS *Q$LBD+Q$LBP>,(SP) ;SET SEARCH NODE + ENABLE INTERRUPTS
215 000572 052716 030000   BIS #Q$CNC,(SP) ;CLEAR ALL SELECTIONS
216 000576 005046   CLR -(SP) ;CLEAR DRIVE AND PULSE
217 000600          CALL CSR1 ;CLEAR CP NO-CLOCK
218                                     ;SET NOTHING
219 000604 032767 000000G-000000G. BIT #RP,BASE
220 000612 001412.   BEQ 2$       ;REPEAT PROMPT?
221 000614          1$: CALL PDATA ;NO
222 000620 103434   BCS LD2X      ;READ DATA FROM COMMAND LINE
223 000622 102433   BVS LD2X
224 000624          CALL LOADCD
225 000630 066767 000000G-000000G. ADD INCVAL,MSTR2
226 000636 000766   BR 1$         ;CR RESPONSE
227                                     ;LOAD ONE WORD
228                                     ;BUMP LOAD ADDRESS
229                                     ;REPEAT
230
231          PROMPT-ONCE THEN-FILL MEMORY
```

```

229
230 000640          ;
231 000644          2$: CALL PDATA          ;PROMPT
232 000646          BCS LD2X          ;END OF MEMORY
233 000650          BVS LD2X          ;<CR> RESPONSE
234 000654 066767 000000G-000000G 3$: CALL LOADCD          ;LOAD 1 WORD
235 000662 026767 000000G-000000G ADD INCVAL,MSTR2          ;ADVANCE ADDRESS
236 000670 101767 000000G-000000G CMP MSTR2,MEND          ;HAS UPPER MEMORY LIMIT BEEN REACHED
237 000672 032767 000000G-000000G BLOS 3$          ;NO CONTINUE
238 000700 001404 000000G-000000G BIT #LOOP,BASE          ;LOOP ON ?
239 000702 016767 000000G-000000G BEQ LD2X          ;NO EXIT
240 000710 000757 000000G-000000G MOV MSTRT,MSTR2          ;REINITIALIZE ADDRESS
241
242 000712 005046          ;
243 000714 012746 176000 LD2X: CLR -(SP)          ;CLEAR NOTHING IN CSR1
244 000720          MOV #0$NCLK,-(SP)          ;SET NO-CLOCK
245 000724 005067 176422 CALL CSR1
246 000730 000167 000340 CLR OR$CR2          ;SET LOAD MODE
247          JMP LD2X          ;EXIT LOAD
248
249          ;
250          ; LOAD HRL BUFFER IN MODULE 'CPBUG'
251 000734          ;
252 000734 016746 000000G LD2HL: MOV HLHIGH,-(SP)          ;SUPPLY UPPER MEMORY LIMIT
253 000744 016746 000000G MOV HLLow,-(SP)          ;SUPPLY LOWER MEMORY LIMIT
254 000750 103551 CALL BUFSM          ;VERIFY CURRENT RANGE
255          BCS LD2X          ;AN ERROR
256 000752 032767 000000G-000000G ;
257 000760 001421          BIT #RP,BASE          ;REPEAT PROMPT
258 000762          BEQ 3$          ;NO FILL WITH ONE VALUE
259 000766 103002 000300 1$: CALL PDATA          ;PROMPT FOR DATA
260 000770 000167 000300 BCC 2$          ;NOT END OF MEMORY
261 000774 102002 000272 2$: JMP LD2X          ;ERROR OR END OF MEMORY
262 000776 000167 000272 BVC 20$          ;NOT <CR> RESPONSE
263          JMP LD2X
264 001002 016700 000000G 20$: MOV MSTR2,R0          ;LOAD MEMORY ADDR (REALLY AN OFFSET)
265 001006 016700 000000G-000000G MOV DATA1,HRL(R0)          ;LOAD HRL TABLE
266 001014 066767 000000G-000000G ADD INCVAL,MSTR2          ;BUMP OFFSET
267 001022 000757 000000G BR 1$          ;PROMPT FOR NEXT VALUE
268
269          ;
270          ; PROMPT ONCE THEN FILL MEMORY
271 001024          ;
272 001030 103002 000236 3$: CALL PDATA          ;PROMPT FOR VALUE
273 001032 000167 000236 BCC 4$          ;NOT END OF MEMORY
274 001036 102002 000230 4$: BVC 40$          ;ERROR ON PROMPT
275 001040 000167 000230 JMP LD2X          ;NOT <CR> RESPONSE
276 001044 016700 000000G 40$: MOV MSTR2,R0          ;LOAD ADDR (REALLY AN OFFSET)
277 001050 016700 000000G-000000G MOV DATA1,HRL(R0)          ;MOVE VALUE TO HRL IN MEMORY
278 001056 066767 000000G-000000G ADD INCVAL,MSTR2          ;BUMP TO NEXT ADDRESS
279 001064 026767 000000G-000000G CMP MSTR2,MEND          ;FINISHED ?
280 001072 101761 000000G-000000G BLOS 4$          ;NO FILL NEXT LOCATION
281 001074 032767 000000G-000000G BIT #LOOP,BASE          ;REPEAT ?
282 001102 001474 000000G-000000G BEQ LD2X          ;NO EXIT
283 001104 016767 000000G-000000G MOV MSTRT,MSTR2          ;RE-INIT START ADDRESS (OFFSET)
284 001112 000751 000000G-000000G BR 4$
285

```



```
286 ;
287 ;
288 ;
289 001114 LD2BL:
290 001114 016746 000000G MOV BLHIGH, -(SP) ;SUPPLY UPPER MEMORY LIMIT
291 001120 016746 000000G MOV BLOW, -(SP) ;SUPPLY LOWER MEMORY LIMIT
292 001124 CALL BUFSM ;VERIFY CURRENT RANGE
293 001130 103461 BCS LD2X ;AN ERROR
294 ;
295 001132 032767 000000G 000000G BIT #RP, BASE ;REPEAT PROMPT
296 001140 001421 BEQ 3$ ;NO, FILL WITH ONE VALUE
297 001142 1$: CALL PDATA ;PROMPT FOR DATA
298 001146 103002 BCC 2$ ;NOT END OF MEMORY
299 001150 000167 000120 JMP LD2X ;ERROR OR END OF MEMORY
300 001154 102002 2$: BVC 20$ ;NOT <CR> RESPONSE
301 001156 000167 000112 JMP LD2X
302 ;
303 001162 016700 000000G 20$: MOV MSTR2, R0 ;LOAD MEMORY ADDR (REALLY AN OFFSET)
304 001166 016760 000000G 000000G MOV DATA1, BCL (R0) ;LOAD BCL TABLE
305 001174 066767 000000G 000000G ADD INCVAL, MSTR2 ;BUMP OFFSET
306 001202 000757 BR 1$ ;PROMPT FOR NEXT VALUE
307 ;
308 ;
309 ;
310 001204 3$: CALL PDATA ;PROMPT FOR VALUE
311 001210 103002 BCC 4$ ;NOT END OF MEMORY
312 001212 000167 000056 JMP LD2X ;ERROR ON PROMPT
313 001216 102002 4$: BVC 40$ ;NOT <CR> RESPONSE
314 001220 000167 000050 JMP LD2X
315 ;
316 001224 016700 000000G 40$: MOV MSTR2, R0 ;LOAD ADDR (REALLY AN OFFSET)
317 001230 016760 000000G 000000G MOV DATA1, BCL (R0) ;MOVE VALUE TO BCL IN MEMORY
318 001236 066767 000000G 000000G ADD INCVAL, MSTR2 ;BUMP TO NEXT ADDRESS
319 001244 026767 000000G 000000G CMP MSTR2, MEND ;FINISHED?
320 001252 101761 BLOS 4$ ;NO, FILL NEXT LOCATION
321 001254 032767 000000G 000000G BIT #LOOP, BASE ;REPEAT?
322 001262 001404 BEQ LD2X ;NO, EXIT
323 001264 016767 000000G 000000G MOV MSTR1, MSTR2 ;RE-INIT START ADDRESS (OFFSET)
324 001272 000751 BR 4$
325 ;
326 ;
327 001274 LD2X:
328 001274 012746 000130 MOV #<CSWRTEN+PLD>, -(SP) ;WRITE DISABL SECT D FW
329 001300 005046 CLR -(SP) ;SET NOTHING FW
330 001302 CALL CSR1 ;
331 001306 042767 000000G 000000G BIC #RP, BASE ;CLEAR PROMPT REPEAT FLAG
332 001314 CALL KILL ;KILL AST
333 001320 000167 000000G JMP CPXX
```

```

335      ;
336      ;
337      ;      LOAD CP CONTROL STORE
338      ;
339      ;
340 001324      LOADCS:
341 001324 316746 000000G      MOV      MSTR2,-(SP)      ;SAVE CURRENT ADDRESS IN STACK
342 001330      CALL      SEQCS      ;SEQUENCE UP TO START ADDRESS
343 001334 012746 000100      MOV      #CSURTEN,-(SP)      ;SET WRITE ENABLE SECT A
344 001340      CALL      CPCRA      ;WRITE CP CONTROL REG
345 001344 016746 000000G      MOV      DATA1,-(SP)      ;SUPPLY SECT A DATA WORD
346 001350      CALL      LBCSC      ;SEND IT TO CP - CLOCK SEQUENCER ONLY
347      ;
348 001354 016746 000000G      MOV      MSTR2,-(SP)      ;SAVE CURRENT ADDRESS IN STACK
349 001360      CALL      SEQCS      ;SEQUENCE UP TO START ADDRESS
350 001364 012746 000110      MOV      #<CSURTEN+PLB>,-(SP)      ;SET WRITE ENABLE SECT B
351 001370      CALL      CPCRA      ;DIRECT CNTL WORD TO CP
352 001374 016746 000000G      MOV      DATA2,-(SP)      ;MOVE DATA WORD TO LOD BUS REG
353 001400      CALL      LBCSC      ;SEND DATA TO CP - CLOCK SEQUENCER ONLY
354      ;
355 001404 016746 000000G      MOV      MSTR2,-(SP)      ;SAVE CURRENT ADDRESS IN STACK
356 001410      CALL      SEQCS      ;SEQUENCE UP TO START ADDRESS
357 001414 012746 000120      MOV      #<CSURTEN+PLC>,-(SP)      ;SET WRITE ENABLE SECT C
358 001420      CALL      CPCRA      ;DIRECT CNTL WORD TO CP
359 001424 016746 000000G      MOV      DATA3,-(SP)      ;MOVE DATA WORD TO LOD BUS REG
360 001430      CALL      LBCSC      ;SEND DATA TO CP - CLOCK SEQUENCER ONLY
361      ;
362 001434 016746 000000G      MOV      MSTR2,-(SP)      ;SAVE CURRENT ADDRESS IN STACK
363 001440      CALL      SEQCS      ;SEQUENCE UP TO START ADDRESS
364 001444 012746 000130      MOV      #<CSURTEN+PLD>,-(SP)      ;WRITE ENABLE SECTION D
365 001450      CALL      CPCRA      ;DIRECT CNTL WORD TO CP
366 001454 016746 000000G      MOV      DATA4,-(SP)      ;MOVE DATA WORD TO LOD BUS REG
367 001460      CALL      LBCSC      ;SEND DATA TO CP - CLOCK SEQUENCER ONLY
368      ;
369 001464 005046      CLR      -(SP)      ;REINHIBIT BRANCH CONTROL REG
370 001466      CALL      CPCR      ;
371 001472 005046      CLR      -(SP)      ;
372 001474      CALL      LBPC      ;SINGLE CLOCK TO REINHIBIT BRANCH REGISTER
373 001500      RETURN
374      ;
375      ;
376      ;      LOAD CP DATA MEMORY
377      ;
378      ;
379 001502      LOADCD:
380 001502 012767 000003 176424      MOV      #Q$LD CD,QR$LBR      ;MOVE ATTN CODE TO LOD BUS REG
381 001510 012767 120100 176422      MOV      #<Q$ATTN+Q$SM+Q$ENOP>,QR$CR2 ;SET ATTN CODE READY
382 001516 016701 176422      1$:      MOV      QR$CR2,R1      ;READ CSR2
383 001522 032701 000100      BIT      #Q$ATTN,R1      ;ATTN CLEAR
384 001526 001373      BNE      1$      ;NO, READ AGAIN
385      ;
386 001530 016767 000000G 176424      MOV      MSTR2,QR$LBR      ;CD MEMORY START ADDRESS
387 001536 012767 120040 176422      MOV      #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
388 001544 016701 176422      2$:      MOV      QR$CR2,R1      ;READ CSR2
389 001550 032701 000040      BIT      #Q$CCCP,R1      ;IS CC TO CP CLEAR
390 001554 001373      BNE      2$      ;NO, READ AGAIN
391

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
392 001556 012767 000001 176424      MOV.    #1,QR$LBR.          ;TRANSFER COUNT = 1 WORD
393 001564 012767 120040 176422      MOV.    *(<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2. ;SET CC TO CP
394 001572 016701 176422      3$:    MOV.    QR$CR2,R1          ;READ CSR2
395 001576 032701 000040          BIT.    #Q$CCCP,R1          ;IS CC TO CP CLEAR
396 001602 001373          BNE.    3$              ;NO, READ AGAIN
397                                     ;
398 001604 012767 000000G 176424      MOV.    #DATA1,QR$LBR.      ;CC MEMORY DATA BUFFER
399 001612 012767 120040 176422      MOV.    *(<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2. ;SET CC TO CP
400                                     ;
401                                     ;
402                                     ;
403 001620          ;
404                                     ;
405 001632          ;
406                                     ;
407                                     ;
408                                     ;
409 001644 012767 100400 176422      MOV.    *(<Q$SM+Q$CHB>,QR$CR2.    ;CLEAR INTERRUPT (USE HIT BUFFER INT)
410 001652 012767 101000 176422      MOV.    *(<Q$SM+Q$REBK>,QR$CR2.    ;RE-ARM
411 001660 012767 160000 176422      MOV.    *(<Q$SM+Q$ENBK+Q$ENOP>,QR$CR2. ;ENABLE
412 001666          RETURN
413                                     ;
414 000001          .END
```

ALUCKE = 000000	BYTE35 = 000043	BYTE87 = 000127	KILL = ***** GX	Q\$CPCC = 000010
ALUOE = 000000	BYTE36 = 000044	BYTE88 = 000130	LBCP = ***** GX	Q\$CP2 = 000260
A01 = 010000	BYTE37 = 000045	BYTE89 = 000131	LBCSC = ***** GX	Q\$CSC = 010000
BASE = ***** GX	BYTE38 = 000046	BYTE9 = 000011	LD2CX = 000712R	002 Q\$CSEL = 000360
BCL = ***** GX	BYTE39 = 000047	BYTE90 = 000132	LD2 = 000020RG	002 Q\$CSET = 000002
BINWD = ***** GX	BYTE4 = 000004	BYTE91 = 000133	LD2BL = 001114RG	002 Q\$CSP = 020000
BITVAL = 000000	BYTE40 = 000050	BYTE92 = 000134	LD2CD = 000472RG	002 Q\$DMA = 000001
JIT0 = 000001	BYTE41 = 000051	BYTE93 = 000135	LD2CS = 000272RG	002 Q\$ENBK = 040000
BIT1 = 000002	BYTE42 = 000052	BYTE94 = 000136	LD2HL = 000734RG	002 Q\$ENOP = 020000
BIT10 = 000200	BYTE43 = 000053	BYTE95 = 000137	LD2LN = 000004 G	Q\$FAL = 000400
BIT11 = 000400	BYTE44 = 000054	BYTE96 = 000140	LD2TBL = 000000RG	002 Q\$FFC = 000045
BIT12 = 010000	BYTE45 = 000055	BYTE97 = 000141	LD2X = 001274R	002 Q\$FO = 000044
BIT13 = 020000	BYTE46 = 000056	BYTE98 = 000142	LOADCD = 001502R	002 Q\$FP = 000046
BIT14 = 040000	BYTE47 = 000057	BYTE99 = 000143	LOADCS = 001324R	002 Q\$HBF = 000002
BIT15 = 100000	BYTE48 = 000060	BYTVAL = 000144	LOC.EN = 000100	Q\$ICP = 000006
BIT2 = 000004	BYTE49 = 000061	CBKALL = 001000	LOC.WA = 040000	Q\$IHB = 000003
BIT3 = 000010	BYTE5 = 000005	CBKCLK = 000400	LOC.WB = 100000	Q\$IHRL = 000002
BIT4 = 000020	BYTE50 = 000062	CDHIGH = ***** GX	LOOP = ***** GX	Q\$IMRP = 000007
BIT5 = 000040	BYTE51 = 000063	CDLOW = ***** GX	MAREN1 = 000001	Q\$LBD = 001000
BIT6 = 000100	BYTE52 = 000064	CNOBRE = 100000	MAREN2 = 000400	Q\$LBDP = 001001
BIT7 = 000200	BYTE53 = 000065	CPCCEN = 010000	MARLOD = 010000	Q\$LBP = 000001
BIT8 = 000400	BYTE54 = 000066	CPCR = ***** GX	MAROUT = 000002	Q\$LCD = 000003
BIT9 = 001000	BYTE55 = 000067	CPCRA = ***** GX	MARLO = 002000	Q\$LDMD = 000004
BLHIGH = ***** GX	BYTE56 = 000070	CPREAD = 040000	MAROU = 000040	Q\$LDPP = 002000
BLOW = ***** GX	BYTE57 = 000071	CPWRT = 020000	MBKALL = 001000	Q\$LHP = 010000
BUFSET = ***** GX	BYTE58 = 000072	CPXX = ***** GX	MBKCLK = 000400	Q\$MNC = 140000
BUFSM = ***** GX	BYTE59 = 000073	CSADPD = 000004	MEND = ***** GX	Q\$MR = 000052
BUFS4 = ***** GX	BYTE6 = 000006	CSEDCI = 100000	MMADRD = 000100	Q\$MRP = 000040
BYTE0 = 000000	BYTE60 = 000074	CSHIGH = ***** GX	MMLEFT = 000002	Q\$MRP2 = 000240
BYTE1 = 000001	BYTE61 = 000075	CSLOW = ***** GX	MMOE = 000004	Q\$MSC = 040000
BYTE10 = 000012	BYTE62 = 000076	CSOE = 000040	MMURTE = 000010	Q\$MSET = 000004
BYTE11 = 000013	BYTE63 = 000077	CSR1 = ***** GX	MNOBRE = 100000	Q\$MSP = 100000
BYTE12 = 000014	BYTE64 = 000100	CSURTE = 000100	MREN1 = 000001	Q\$NCLK = 176000
BYTE13 = 000015	BYTE65 = 000101	DATA1 = ***** GX	MREN2 = 020000	Q\$PP = 000100
BYTE14 = 000016	BYTE66 = 000102	DATA2 = ***** GX	MRPCR = ***** GX	Q\$PPSW = 000320
BYTE15 = 000017	BYTE67 = 000103	DATA3 = ***** GX	MSTRT = ***** GX	Q\$PP2 = 000300
BYTE16 = 000020	BYTE68 = 000104	DATA4 = ***** GX	MSTR2 = ***** GX	Q\$QHLT = 000013
BYTE17 = 000021	BYTE69 = 000105	DBR.RD = 000001	MSYN = 000040	Q\$QL = 000043
BYTE18 = 000022	BYTE7 = 000007	DB\$CPP = 001457	N = 000144	Q\$QLA = 000053
BYTE19 = 000023	BYTE70 = 000106	DB\$SPT = 000026	PACK = ***** GX	Q\$QLB = 000054
BYTE2 = 000002	BYTE71 = 000107	DB\$TPC = 000023	PDATA = ***** GX	Q\$QLR = 000001
BYTE20 = 000024	BYTE72 = 000110	DISPGS = 100000	PLB = 000010	Q\$QW = 000042
BYTE21 = 000025	BYTE73 = 000111	DMAWR = 000005	PLC = 000020	Q\$RDCD = 000005
BYTE22 = 000026	BYTE74 = 000112	DMARRD = 000003	PLD = 000030	Q\$RDMD = 000006
BYTE23 = 000027	BYTE75 = 000113	DMARWR = 000004	PLRWR = 000200	Q\$REBK = 001000
BYTE24 = 000030	BYTE76 = 000114	EFN.3 = ***** GX	PLR.EN = 000200	Q\$RNC = 006000
BYTE25 = 000031	BYTE77 = 000115	ENBR = 010000	Q\$CR1 = 176420	Q\$RSC = 004000
BYTE26 = 000032	BYTE78 = 000116	ERR11 = ***** GX	Q\$CR2 = 176422	Q\$RSET = 000010
BYTE27 = 000033	BYTE79 = 000117	ERR4 = ***** GX	Q\$ELBR = 176424	Q\$SM = 100000
BYTE28 = 000034	BYTE8 = 000010	ERR5 = ***** GX	Q\$ATTN = 000100	Q\$SP = 000120
BYTE29 = 000035	BYTE80 = 000120	ERR7 = ***** GX	Q\$BCL = 000001	Q\$SP2 = 000340
BYTE3 = 000003	BYTE81 = 000121	FIND = ***** GX	Q\$CCCP = 000040	RQO.EN = 000200
BYTE30 = 000036	BYTE82 = 000122	HANG = ***** GX	Q\$CHB = 000400	RQO.VA = 020000
BYTE31 = 000037	BYTE83 = 000123	HLHIGH = ***** GX	Q\$CHRL = 000200	RP = ***** GX
BYTE32 = 000040	BYTE84 = 000124	HLLOW = ***** GX	Q\$CLR = 000040	RTNPT = ***** GX
BYTE33 = 000041	BYTE85 = 000125	HRL = ***** GX	Q\$CNC = 030000	SCAN = ***** GX
BYTE34 = 000042	BYTE86 = 000126	INCVL = ***** GX	Q\$CP = 000060	SEQCS = ***** GX

CPLD...MACRO.M1110 27-MAR-80 14:43 PAGE 6-3
SYMBOL TABLE

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

SEQMM= .:***** GX.	T\$FSAA= .000000	WORD12= .000030	WORD42= .000124	WORD72= .000220
SEQCI= .000010	T\$FSAB= .000004	WORD13= .000032	WORD43= .000126	WORD73= .000222
S\$CLR= .000000	T\$FSAC= .000014	WORD14= .000034	WORD44= .000130	WORD74= .000224
S\$LA= .000001	T\$FSB2= .000010	WORD15= .000036	WORD45= .000132	WORD75= .000226
S\$QB= .000005	T\$IB= .000026	WORD16= .000040	WORD46= .000134	WORD76= .000230
S\$QR= .000006	T\$IBAR= .000024	WORD17= .000042	WORD47= .000136	WORD77= .000232
S\$QX= .000004	T\$IBE= .020000	WORD18= .000044	WORD48= .000140	WORD78= .000234
S\$SR= .000007	T\$IBF= .040000	WORD19= .000046	WORD49= .000142	WORD79= .000236
S\$S1= .000010	T\$ICD= .000040	WORD2= .000004	WORD5= .000012	WORD8= .000020
S\$S2= .000014	T\$MODE= .004000	WORD20= .000050	WORD50= .000144	WORD80= .000240
TD\$CTR= .176370	T\$OB= .000036	WORD21= .000052	WORD51= .000146	WORD81= .000242
TD\$CTW= .176360	T\$OBE= .004000	WORD22= .000054	WORD52= .000150	WORD82= .000244
TD\$INL= .004000	T\$OBF= .010000	WORD23= .000056	WORD53= .000152	WORD83= .000246
TD\$MEM= .000270	T\$OBRA= .000034	WORD24= .000060	WORD54= .000154	WORD84= .000250
TD\$OAR= .176344	T\$OBWA= .000032	WORD25= .000062	WORD55= .000156	WORD85= .000252
TD\$OTR= .176346	T\$OUTA= .100000	WORD26= .000064	WORD56= .000160	WORD86= .000254
TD\$QRD= .000274	T\$RBD0= .000200	WORD27= .000066	WORD57= .000162	WORD87= .000256
TD\$SW= .176376	T\$RNB= .000040	WORD28= .000070	WORD58= .000164	WORD88= .000260
TD\$TAR= .176372	T\$RSET= .040000	WORD29= .000072	WORD59= .000166	WORD89= .000262
TD\$TAU= .176362	T\$SC= .000022	WORD3= .000006	WORD6= .000014	WORD9= .000022
TD\$TDR= .176374	T\$SCLK= .020000	WORD30= .000074	WORD60= .000170	WORD90= .000264
TD\$TDW= .176364	T\$SEG1= .000000	WORD31= .000076	WORD61= .000172	WORD91= .000266
T\$AD= .000020	T\$SEG2= .000001	WORD32= .000100	WORD62= .000174	WORD92= .000270
T\$BA= .000002	T\$SEG3= .000002	WORD33= .000102	WORD63= .000176	WORD93= .000272
T\$BI= .000010	T\$SO= .000001	WORD34= .000104	WORD64= .000200	WORD94= .000274
T\$BSO= .100000	T\$UBUS= .100000	WORD35= .000106	WORD65= .000202	WORD95= .000276
T\$BT= .000020	T\$1CLK= .000400	WORD36= .000110	WORD66= .000204	WORD96= .000300
T\$BTAR= .000030	T\$BEN= .000020	WORD37= .000112	WORD67= .000206	WORD97= .000302
T\$BTD= .002000	UBD, IN= .000020	WORD38= .000114	WORD68= .000210	WORD98= .000304
T\$CD= .000100	WORD0= .000000	WORD39= .000116	WORD69= .000212	WORD99= .000306
T\$CLK= .002000	WORD1= .000002	WORD4= .000010	WORD7= .000016	WORDVAL= .000310
T\$DISK= .000200	WORD10= .000024	WORD40= .000120	WORD70= .000214	XTREAD= .001000
T\$DRD= .000004	WORD11= .000026	WORD41= .000122	WORD71= .000216	XTWRITE= .000400
T\$MEM= .010000				

. ABS. 000000 000
000000 001
CPLD 001670 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3677 WORDS (15 PAGES)
DYNAMIC MEMORY: 4916 WORDS (18 PAGES)
ELAPSED TIME: 00:00:51
CPLD, CPLD-SP=[20, 1]IM, [20, 1]CPLD

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

1      .TITLE CPPR-
2      .PSECT CPPR-
3
4
5
6      HARDWARE QUERY RESOLVER "MANUAL" DEBUGGING AIDS
7      CONTROL PROCESSOR TEST ROUTINES
8      ***** PROTOTYPE VERSION *****
9
10     COMMANDS:
11     PR      PRINT CP MEMORY CONTENTS
12
13     ONCE A COMMAND HAS BEEN EXECUTED (OR AN ERROR ENCOUNTERED)
14     THIS MODULE RETURNS CONTROL TO THE MODULE CP AT LOCATION
15     'CPXX'.
16
17     *****
18
19     NOTE:
20     THE COMMAND TO PRINT CP DATA MEMORY CANNOT BE USED
21     UNLESS MRP MICROPROGRAM MEMORY AND CP CONTROL STORE
22     HAVE BEEN LOADED VIA THE "LOADER" PROGRAM.
23     *****
24
25
26
27     .MCALL WTSE$S,CLEF$S
28
29
30     TABLE OF VALID CP MEMORY MNEMONICS AND ADDRESSES OF ASSOCIATED
31     ROUTINES.
32
33     000000      103      123
34     000000      000312
35     000004      103      104
36     000006      000632
37     000002
38
39     PR2TBL:
40     .ASCII /CS/      :PRINT MICROPGM MEMORY
41     .WORD PR2CS
42     .ASCII /CD/      :PRINT DATA MEMORY
43     .WORD PR2CD
44     PR2LN == <.-PR2TBL>/4
45
46     PRINT BUFFER
47
48     PB2TBL:
49     .ASCII /HB/      :PRINT HIT BUFFER
50     .WORD PB2HB
51     .ASCII /HL/      :PRINT HRL
52     .WORD PB2HL
53     .ASCII /BL/      :PRINT BCL
54     .WORD PB2BL
55     PB2LN == <.-PB2TBL>/4
56
57     PRINT
58     PERFORM THIRD LEVEL PARSING
59     EG. IN THE COMMAND:
60     PR>PR CD 0
61     PARSE THE "CD"

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

58
59 000024      ;
60 000024      PR2::
61 000030      103004      CALL    FIND          ;LOCATE MEMORY MNEMONIC IN COMMAND LINE
62 000032      BCC        1$          ;OK, CONTINUE
63 000036      000167 001164      CALL    ERR4          ;MISSING OPERAND
64      JMP        PR2X              ;EXIT
65 000042      012700 000002      1$: MOV    #PR2LN,R0      ;NUMBER OF TABLE ENTRIES
66 000046      012702 000000      MOV    #PR2TBL,R2      ;POINT TO TABLE
67 000052      CALL    SCAN          ;MATCH AGAINST COMMAND LINE
68 000056      103004      BCC        2$          ;MATCH WAS MADE
69 000060      CALL    ERR7          ;INVALID MEMORY MNEMONIC
70 000064      000167 001136      JMP    PR2X
71
72      ;
73      ;
74      ;
75      ;
76      ;
77      ;
78      ;
79      ;
80      ;
81      ;
82      ;
83      ;
84      ;
85      ;
86      ;
87      ;
88      ;
89      ;
90 000070      010167 000000G      2$: MOV    R1,RTNPT      ;SAVE POINTER
91 000074      103004      CALL    FIND          ;LOCATE NON-BLANK IN COMMAND LINE
92 000100      BCC        3$          ;OK, CONTINUE
93 000102      000167 001114      CALL    ERR4          ;MISSING OPERAND
94 000112      JMP        PR2X              ;EXIT
95 000116      103004      3$: CALL    PACK          ;CONVERT COMMAND LINE VALUE TO BINARY
96 000120      000120      BCC        4$          ;CONVERSION SUCCESSFUL
97 000124      000167 001076      CALL    ERR5          ;INVALID NUMERIC VALUE
98 000128      JMP        PR2X
99
100 000130      016767 000000G 000000G 4$: MOV    BINWD,MSTRT      ;SAVE PRINT START ADDRESS
101 000136      016767 000000G 000000G      MOV    BINWD,MSTR2      ;SAVE IT TWICE
102 000144      012767 177777 000000G      MOV    *-1,MEND      ;INIT END ADDRESS
103
104      ;
105      ;
106      ;
107      ;
108 000152      CALL    FIND          ;SCAN COMMAND LINE
109 000156      103004      BCC        5$          ;SOMETHING THERE
110 000160      016767 000000G 000000G      MOV    MSTR,MEND      ;SET END ADDR = START ADDR
111 000166      000445      BR         9$          ;JUMP TO RTN
112
113 000170      122711 000114      5$: CMPB   *-L,(R1)      ;LOOP INDICATOR
114 000174      001011      BNE        6$          ;NO, MUST BE UPPER ADDRESS

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

115 000176 052767 000000G-000000G- BIS- #OUT,BASE- ;SET FLAG FOR OUTPUT CONTROL
116 000204 016767 000000G-000000G- MOV- MSTRT,MEND- ;SET END ADDR = START ADDR
117 000212- CALL- HANG ;HOW TO STOP LOOP
118 000216 000431 BR- 9$ ;JUMP TO RTN
119-
120 000220 6$: CALL- PACK ;CONVERT UPPER ADDRESS
121 000224 103004 BCC- 7$ ;OK, CONTINUE
122-000226 CALL- ERR5 ;INVALID NUMERIC
123 000232- 000167 000770 JMP- PR2X ;EXIT
124-
125-
126-
127-
128-000236 016767 000000G-000000G-7$: MOV- BINWD,MEND- ;SET UP END ADDRESS
129 000244 CALL- FIND ;CHECK FOR LOOP INDICATOR
130 000250 103414 BCS- 9$ ;NO LOOP
131 000252- 122711 000114 CMPB- #L,(R1) ;CORRECT LOOP INDICATOR
132-000256 001404 BEQ- 8$ ;YES
133 000260 CALL- ERR11
134 000264 000167 000736 JMP- PR2X ;NO
135 000270 052767 000000G-000000G-8$: BIS- #OUT,BASE- ;SET OUTPUT CONTROL
136 000276 CALL- HANG ;HOW TO STOP LOOP
137-
138 000302- 016701 000000G- 9$: MOV- RTNPT,R1 ;POINT TO ROUTINE
139 000306 000171 000000 JMP- @ (R1) ;JUMP TO ROUTINE
140-
141-
142-
143-
144-
145 000312- PR2CS::
146 000312- 005046 CLR- -(SP) ;CLEAR NOTHING
147 000314 012746 000002 MOV- #Q$CSET,-(SP) ;CP RESET
148 000320 CALL- CSR1 ;DO IT
149 000324 012746 000002 MOV- #Q$CSET,-(SP) ;CLEAR CP RESET
150 000330 005046 CLR- -(SP) ;SET NOTHING
151 000332- CALL- CSR1
152-
153 000336 016746 000000G- MOV- CSHIGH,-(SP) ;SUPPLY UPPER MEMORY LIMIT
154 000342- 016746 000000G- MOV- CSLW,-(SP) ;LOWER LIMIT
155 000346 CALL- BUF54 ;PREPARE FOR LOAD
156 000352- 103002- BCC- 1$ ;NO ERROR
157 000354 000167 000646 JMP- PR2X
158-
159 000360 316746 000000G- 1$: MOV- MSTR2,-(SP) ;SEND CURRENT ADDRESS
160 000364 CALL- SEQCS ;SEQUENCE UP TO PRINT START ADDRESS
161 000370 012746 000040 MOV- #CSOE,-(SP) ;SELECT CNTL STORE SECT A
162-000374 CALL- CPR ;SET CP CNTL REG
163 000400 005046 CLR- -(SP)
164 000402 CALL- LBCP ;CLOCK TO RESET BR REG FF
165 000406 CALL- CPLB ;REQUEST CP TO LOD BUS
166 000412- 012667 000000G- MOV- (SP)+,DATA1 ;SAVE FOR PRINTING
167-
168 000416 016746 000000G- MOV- MSTR2,-(SP) ;SEND CURRENT ADDRESS
169 000422- CALL- SEQCS ;SEQUENCE UP TO PRINT START ADDRESS
170 000426 012746 000050 MOV- #<CSOE+PLB>,-(SP) ;SELECT CNTL STORE SECT B
171 000432- CALL- CPR ;SET CP CNTL REG

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3


```
172 000436 005046 CLR - (SP)
173 000440 CALL LBCL :CLOCK TO REINHIBIT BR
174 000444 CALL CPLB :REQUEST CP TO LOD BUS
175 000450 012667 000000G MOV (SP)+, DATA2 :SAVE FOR PRINTING
176
177 000454 016746 000000G MOV MSTR2, - (SP) :SEND CURRENT ADDRESS
178 000460 CALL SEQCS :SEQUENCE UP TO PRINT START ADDRESS
179 000464 012746 000060 MOV *(CSOE+PLD), - (SP) :SELECT CNTL STORE SECT C
180 000470 CALL CPCR :SET CP CNTL REG
181 000474 005046 CLR - (SP)
182 000476 CALL LBCL :CLOCK TO REINHIBIT BR
183 000502 CALL CPLB :REQUEST CP TO LOD BUS
184 000506 012667 000000G MOV (SP)+, DATA3 :SAVE FOR PRINTING
185
186 000512 016746 000000G MOV MSTR2, - (SP) :SEND CURRENT ADDRESS
187 000516 CALL SEQCS :SEQUENCE UP TO PRINT START ADDRESS
188 000522 012746 000070 MOV *(CSOE+PLD), - (SP) :SELECT CNTL STORE SECT D
189 000526 CALL CPCR :SET CP CNTL REG
190 000532 005046 CLR - (SP)
191 000534 CALL LBCL :CLOCK TO REINHIBIT BR
192 000540 CALL CPLB :REQUEST CP TO LOD BUS
193 000544 012667 000000G MOV (SP)+, DATA4 :SAVE FOR PRINTING
194 000550 005046 CLR - (SP) :CLEAR CP CONTROL REG
195 000552 CALL CPCR
196
197 000556 CALL PRDATA :PRINT MEMORY CONTENTS
198 000562 103002 BCC 100$ :NOT END OF MEMORY
199 000564 000167 000436 JMP PR2X :END OF MEMORY
200
201 000570 066767 000000G-000000G-100$ ADD INCVAL, MSTR2 :ADVANCE ADDRESS
202 000576 026767 000000G-000000G CMP MSTR2, MEND :HAS UPPER MEMORY LIMIT BEEN REACHED
203 000604 101665 BLOS 1$ :NO, CONTINUE
204 000606 032767 000000G-000000G BIT #LOOP, BASE :LOOP ON ?
205 000614 001002 BNE 200$ :YES, CONTINUE
206 000616 000167 000404 JMP PR2X :NO, EXIT
207 000622 016767 000000G-000000G-200$ MOV MSTR2, MSTR2 :INIT START ADDRESS
208 000630 000653 BR 1$ :AND REPEAT PRINT
209
210
211
212 PRINT FROM DATA MEMORY
213 000632 PR2GD::
214 000632 016746 000000G MOV CDHIGH, - (SP) :SUPPLY MEMORY UPPER LIMIT
215 000636 016746 000000G MOV CDLOW, - (SP) :LOWER LIMIT
216 000642 CALL BUFSET :PREPARE FOR LOAD
217 000646 103002 BCC 100$ :OK, CONTINUE
218 000650 000167 000352 JMP PR2X
219
220 000654 005046 100$ CLR - (SP) :START MICROCODE AT 0
221 000656 CALL SEQCS
222
223 000662 005046 CLR - (SP) :REINHIBIT BRANCH CONTROL REG
224 000664 CALL CPCR
225 000670 005046 CLR - (SP)
226 000672 CALL LBCL :SINGLE CLOCK TO REINHIBIT BRANCH REGISTER
227 000676 012746 000377 MOV *(377, - (SP) :SET MRP MICRO ADDRESS = X'FF' (JUMP SELF)
228 000702 CALL SEQMM
```

```
229      :
230 000706 005046      GLR      -(SP)      ;REINHIBIT BRANCH CONTROL REG.
231 000710      CALL     MRPCR      ;
232 000714 005046      CLR      -(SP)      ;
233 000716      CALL     LBMRP      ;SINGLE CLOCK TO REINHIBIT BRANCH REGISTER
234 000722 012767 001000 176422 MOV      *Q$REBK,QR$CR2      ;RE-ARM INTERRUPTS
235 000730 012767 120000 176422 MOV      *Q$SM+Q$ENOP>,QR$CR2 ;SET SEARCH MODE + ENABLE INTERRUPTS
236 000736 012746 000360      MOV      *Q$CSEL,-(SP)      ;CLEAR ALL SELECTIONS
237 000742 052716 001001      BIS      *Q$LBD+Q$LBP>,(SP) ;CLEAR DRIVE AND PULSE
238 000746 052716 030000      BIS      *Q$CNC,(SP)      ;CLEAR CP NO-CLOCK
239 000752 005046      CLR      -(SP)      ;SET NOTHING
240 000754      CALL     CSR1
241      :
242 000760 012767 000005 176424 PRCD: MOV      *Q$RDCD,QR$LBR      ;MOVE ATTN CODE TO LOD-BUS REG
243 000766 012767 120100 176422 MOV      *Q$ATTN+Q$SM+Q$ENOP>,QR$CR2 ;SET ATTN CODE READY
244 000774 016701 176422 1$: MOV      QR$CR2,R1      ;READ CSR2
245 001000 032701 000100      BIT      *Q$ATTN,R1      ;ATTN CLEAR
246 001004 001373      BNE      1$      ;NO, READ AGAIN
247      :
248 001006 016767 000000G 176424 MOV      MSTR2,QR$LBR      ;CD MEMORY START ADDRESS
249 001014 012767 120040 176422 MOV      *Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
250 001022 016701 176422 2$: MOV      QR$CR2,R1      ;READ CSR2
251 001026 032701 000040      BIT      *Q$CCCP,R1      ;IS CC TO CP CLEAR
252 001032 001373      BNE      2$      ;NO, READ AGAIN
253      :
254 001034 012767 000001 176424 MOV      #1,QR$LBR      ;TRANSFER COUNT = 1 WORD
255 001042 012767 120040 176422 MOV      *Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
256 001050 016701 176422 3$: MOV      QR$CR2,R1      ;READ CSR2
257 001054 032701 000040      BIT      *Q$CCCP,R1      ;IS CC TO CP CLEAR
258 001060 001373      BNE      3$      ;NO, READ AGAIN
259      :
260 001062 012767 000000G 176424 MOV      #DATA1,QR$LBR      ;CC MEMORY DATA BUFFER
261 001070 012767 120040 176422 MOV      *Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
262      :
263      :
264      :
265 001076      WAIT     FOR INTERRUPT FROM CP
266      :
267 001110      CLEF$S  #EFN.3
268      :
269      :
270      :
271 001122 012767 100400 176422 MOV      *Q$SM+Q$CHB>,QR$CR2      ;CLEAR INTERRUPT (USE HIT BUFFER INT)
272 001130 012767 101000 176422 MOV      *Q$SM+Q$REBK>,QR$CR2      ;RE-ARM
273 001136 012767 160000 176422 MOV      *Q$SM+Q$ENBK+Q$ENOP>,QR$CR2 ;ENABLE
274      :
275 001144      CALL     PRDATA      ;PRINT MEMORY CONTENTS
276 001150 103417      BCS      PRCDX      ;END OF MEMORY, EXIT
277      :
278 001152 066767 000000G 000000G ADD      INCVAL,MSTR2      ;ADVANCE ADDRESS
279 001160 026767 000000G 000000G CMP      MSTR2,MEND      ;HAS UPPER MEMORY LIMIT BEEN REACHED
280 001166 101674      BLOS     PRCD      ;NO, CONTINUE
281 001170 032767 000000G 000000G BIT      #LOOP,BASE      ;LOOP ON?
282 001176 001404      BEQ      PRCDX      ;NO, CONTINUE
283 001200 016767 000000G 000000G MOV      MSTR1,MSTR2      ;INIT START ADDRESS
284 001206 000664      BR       PRCD      ;AND REPEAT PRINT
285
```

286	001210	005046		PRCDX:	CLR	-(SP)		;CLEAR NOTHING IN CSR1
287	001212	012746	176000		MOV	#0\$NCLK, -(SP)		;SET NO-CLOCKS
288	001216				CALL	CSR1		
289	001222	005067	176422		CLR	QR\$CR2		;SET LOAD MODE
290								
291	001226			PR2X:				
292	001226	042767	0000000 0000000		BIC	*(<ONCE+OUT>,BASE		;CLEAR PRINT CONTROL FLAGS
293	001234				CALL	KILL		;KILL AST
294	001240	000167	0000000		JMP	CPXX		

```

296      ;
297      ;
298      ; PRINT-BUFFER COMMAND.
299      ;
300      ;
301      PB2::
302      001244      CALL    FIND      ;LOCATE NON-BLANK IN COMMAND-LINE
303      001250      BCC     1$      ;OK, CONTINUE
304      001252      CALL    ERR4     ;MISSING OPERAND
305      001256      000167 000272    JMP     PB2X      ;EXIT
306      ;
307      001262      012700 000003    1$: MOV     #PB2LN,R0      ;NUMBER OF TABLE ENTRIES
308      001266      012702 000010    MOV     #PB2TBL,R2     ;POINT TO TABLE
309      001272      CALL    SCAN     ;MATCH AGAINST COMMAND-LINE
310      001276      103004      BCC     2$      ;MATCH WAS MADE
311      001300      CALL    ERR7     ;INVALID MEMORY MNEMONIC
312      001304      000167 000244    JMP     PB2X
313      ;
314      001310      000171 000000    2$: JMP     @ (R1)      ;JUMP TO ROUTINE
315      ;
316      ;
317      ; PRINT-HIT-BUFFER.
318      ;
319      ;
320      PB2HB::
321      001314      CALL    HANG      ;HOW TO STOP PRINT
322      001320      012703 000000G    MOV     #HLB,R3      ;R3 -> HIT LIST-BUFFER
323      001324      012700 000200    MOV     #120,R0      ;NUMBER OF WORDS IN BUFFER
324      001330      012705 000000G    1$: MOV     #PRINT,R5     ;R5 -> PRINT-LINE
325      001334      012702 000014    MOV     #12,R2      ;NUMBER OF WORDS PER PRINT-LINE
326      001340      012301      2$: MOV     (R3)+,R1     ;LOAD WORD TO PRINT INTO R1
327      001342      CALL    UNPK     ;CONVERT TO ASCII-HEX
328      001346      005300      DEC     R0      ;FINISHED WITH TABLE ?
329      001350      001413      BEQ     4$      ;YES, EXIT
330      001352      005302      DEC     R2      ;FINISHED WITH CURRENT PRINT-LINE ?
331      001354      001402      BEQ     3$      ;YES, WRITE TO CONSOLE
332      001356      005205      INC     R5      ;BUMP PRINT-LINE POINTER
333      001360      000767      BR      2$      ;GET NEXT WORD
334      ;
335      001362      3$: CALL    CONSOL      ;WRITE TO TERMINAL
336      001366      032767 000000G-000000G    BIT     #LOOP,BASE ;CONTINUE WRITING
337      001374      001467      BEQ     PB2X      ;NO, EXIT
338      001376      000754      BR      1$      ;INITIALIZE NEXT PRINT-LINE
339      001400      4$: CALL    CONSOL      ;WRITE TO TERMINAL
340      001404      000463      BR      PB2X
341      ;
342      ;
343      ; PRINT-OUT-HRL-BUFFER.
344      ;
345      ;
346      PB2HL::
347      001406      CALL    HANG      ;HOW TO STOP PRINT
348      001412      012703 000000G    MOV     #HRL,R3      ;R3 -> HRL BUFFER
349      001416      012700 000200    MOV     #120,R0      ;NUMBER OF WORDS IN BUFFER
350      001422      012705 000000G    1$: MOV     #PRINT,R5     ;R5 -> PRINT-LINE
351      001426      012702 000014    MOV     #12,R2      ;NUMBER OF WORDS PER PRINT-LINE
352      001432      012301      2$: MOV     (R3)+,R1     ;LOAD WORD TO PRINT INTO R1

```

```

353 001434          CALL    UNPK          ;CONVERT TO ASCII-HEX.
354 001440 005300    DEC     R0            ;FINISHED WITH TABLE ?
355 001442 001413    BEQ     4$           ;YES, EXIT
356 001444 005302    DEC     R2            ;FINISHED WITH CURRENT PRINT LINE ?
357 001446 001402    BEQ     3$           ;YES, WRITE TO CONSOLE
358 001450 005205    INC     R5            ;BUMP PRINT LINE POINTER
359 001452 000767    BR      2$           ;GET NEXT WORD
360
361 001454          3$: CALL    CONSOL      ;WRITE TO TERMINAL
362 001456 032767 000000G 000000G    BIT    #LOOP,BASE ;CONTINUE WRITING
363 001466 001432    BEQ     PB2X         ;NO, EXIT
364 001470 000754    BR      1$           ;INITIALIZE NEXT PRINT LINE
365 001472          4$: CALL    CONSOL      ;WRITE TO TERMINAL
366 001476 300426    BR      PB2X
367
368
369          PRINT OUT BCL
370
371
372 001500          PB2BL: MOV     #BCL,R3      ;R3 -> BCL BUFFER
373 001500 012703 000000G    MOV     #13,R0     ;NUMBER OF WORDS IN BUFFER
374 001504 012700 000015    MOV     #PRINT,R5   ;R5 -> PRINT LINE
375 001510 012705 000000G    1$: MOV     #7,R2     ;NUMBER OF WORDS PER PRINT LINE
376 001514 012702 000007    2$: MOV     (R3)+,R1  ;LOAD WORD TO PRINT INTO R1
377 001520 012301          CALL    UNPK          ;CONVERT TO ASCII-HEX
378 001522          DEC     R0            ;FINISHED WITH TABLE ?
379 001526 005300    BEQ     4$           ;YES, EXIT
380 001530 001407    DEC     R2            ;FINISHED WITH CURRENT PRINT LINE ?
381 001532 005302    BEQ     3$           ;YES, WRITE TO CONSOLE
382 001534 001402    INC     R5            ;BUMP PRINT LINE POINTER
383 001536 005205    BR      2$           ;GET NEXT WORD
384 001540 000767
385
386 001542          3$: CALL    CONSOL      ;WRITE TO TERMINAL
387 001546 000760    BR      1$           ;INITIALIZE NEXT PRINT LINE
388 001550          4$: CALL    CONSOL      ;WRITE TO TERMINAL
389
390 001554          PB2X: CALL    KILL        ;KILL AST
391 001554          JMP     CPXX
392 001560 000167 000000G
393
394          .END

```

ALUCKE = 040000	BYTE38 = 000046	BYTE9 = 000011	LOC.EN = 000100	Q\$CHRL = 000200
ALUOE = 004000	BYTE39 = 000047	BYTE90 = 000132	LOC.WA = 040000	Q\$CLR = 000040
A01 = 010000	BYTE4 = 000004	BYTE91 = 000133	LOC.WB = 100000	Q\$CNC = 030000
BASE = ***** GX	BYTE40 = 000050	BYTE92 = 000134	LOOP = ***** GX	Q\$CP = 000060
BCL = ***** GX	BYTE41 = 000051	BYTE93 = 000135	MAREN1 = 000001	Q\$CPCC = 000010
BINWD = ***** GX	BYTE42 = 000052	BYTE94 = 000136	MAREN2 = 004000	Q\$CP2 = 000260
BITVAL = 000000	BYTE43 = 000053	BYTE95 = 000137	MARLOD = 010000	Q\$CSC = 010000
BIT0 = 000001	BYTE44 = 000054	BYTE96 = 000140	MAROUT = 000002	Q\$CSEL = 000360
BIT1 = 000002	BYTE45 = 000055	BYTE97 = 000141	MAR.LO = 002000	Q\$CSET = 000002
BIT10 = 002000	BYTE46 = 000056	BYTE98 = 000142	MAR.OU = 000040	Q\$CSP = 020000
BIT11 = 004000	BYTE47 = 000057	BYTE99 = 000143	MBKALL = 001000	Q\$DMA = 000001
BIT12 = 010000	BYTE48 = 000060	BYTVAL = 000144	MBKCLK = 000400	Q\$ENBK = 040000
BIT13 = 020000	BYTE49 = 000061	CBKALL = 001000	MEND = ***** GX	Q\$ENOP = 020000
BIT14 = 040000	BYTE5 = 000005	CBKCLK = 000400	MMADRD = 000100	Q\$FAL = 004000
BIT15 = 100000	BYTE50 = 000062	CDHIGH = ***** GX	MMLEFT = 000002	Q\$FC = 000045
BIT2 = 000004	BYTE51 = 000063	CDLOW = ***** GX	MMOE = 000004	Q\$FO = 000044
BIT3 = 000010	BYTE52 = 000064	CNOBRE = 100000	MMURTE = 000010	Q\$FP = 000046
BIT4 = 000020	BYTE53 = 000065	CONSOL = ***** GX	MNOBRE = 100000	Q\$HBF = 000002
BIT5 = 000040	BYTE54 = 000066	CPCCEN = 010000	MREN1 = 000001	Q\$ICP = 000006
BIT6 = 000100	BYTE55 = 000067	CPCR = ***** GX	MREN2 = 020000	Q\$IH = 000003
BIT7 = 000200	BYTE56 = 000070	CPLB = ***** GX	MRPCR = ***** GX	Q\$IHRL = 000002
BIT8 = 000400	BYTE57 = 000071	CPREAD = 040000	MSTRT = ***** GX	Q\$IMRP = 000007
BIT9 = 001000	BYTE58 = 000072	CPURTE = 020000	MSTR2 = ***** GX	Q\$LBD = 001000
BUFSET = ***** GX	BYTE59 = 000073	CPXX = ***** GX	MSYN = 000040	Q\$LBDF = 001001
BUFS4 = ***** GX	BYTE6 = 000006	CSADRD = 000004	N = 000144	Q\$LBP = 000001
BYTE0 = 000000	BYTE60 = 000074	CSEOC I = 100000	ONCE = ***** GX	Q\$LDOD = 000003
BYTE1 = 000001	BYTE61 = 000075	CSHIGH = ***** GX	OUT = ***** GX	Q\$LDOD = 000004
BYTE10 = 000012	BYTE62 = 000076	CSLOW = ***** GX	PACK = ***** GX	Q\$LDPP = 002000
BYTE11 = 000013	BYTE63 = 000077	CSOE = 000040	PB2 = 001244RG	002 Q\$LHP = 010000
BYTE12 = 000014	BYTE64 = 000100	CSR1 = ***** GX	PB2BL = 001500RG	002 Q\$MNC = 140000
BYTE13 = 000015	BYTE65 = 000101	CSWRTE = 000100	PB2HB = 001314RG	002 Q\$MR = 000052
BYTE14 = 000016	BYTE66 = 000102	DATA1 = ***** GX	PB2HL = 001406RG	002 Q\$MRP = 000040
BYTE15 = 000017	BYTE67 = 000103	DATA2 = ***** GX	PB2LN = 000003 G	Q\$MRP2 = 000240
BYTE16 = 000020	BYTE68 = 000104	DATA3 = ***** GX	PB2TBL = 000010RG	002 Q\$MSP = 040000
BYTE17 = 000021	BYTE69 = 000105	DATA4 = ***** GX	PB2X = 001554R	002 Q\$MSET = 000004
BYTE18 = 000022	BYTE7 = 000007	DBR.RD = 000001	PLB = 000010	Q\$MSP = 100000
BYTE19 = 000023	BYTE70 = 000106	DB\$CPP = 001457	PLC = 000020	Q\$NCLK = 176000
BYTE2 = 000002	BYTE71 = 000107	DB\$PTC = 000026	PLD = 000030	Q\$PP = 000100
BYTE20 = 000024	BYTE72 = 000110	DB\$TPC = 000023	PLRWR = 000200	Q\$PPSW = 000320
BYTE21 = 000025	BYTE73 = 000111	DISPGS = 100000	PLR.EN = 000200	Q\$PP2 = 000300
BYTE22 = 000026	BYTE74 = 000112	DMANWR = 000005	PRCD = 000760R	002 Q\$OHLT = 000013
BYTE23 = 000027	BYTE75 = 000113	DMARRD = 000003	PRCDX = 001210R	002 Q\$QL = 000043
BYTE24 = 000030	BYTE76 = 000114	DMARWR = 000004	PRDATA = ***** GX	Q\$QLA = 000053
BYTE25 = 000031	BYTE77 = 000115	EFN.3 = ***** GX	PRINT = ***** GX	Q\$QLB = 000054
BYTE26 = 000032	BYTE78 = 000116	ENBR = 010000	PR2 = 000024RG	002 Q\$CLR = 000001
BYTE27 = 000033	BYTE79 = 000117	ERR11 = ***** GX	PR2CD = 000532RG	002 Q\$QW = 000042
BYTE28 = 000034	BYTE8 = 000010	ERR4 = ***** GX	PR2CS = 000312RG	002 Q\$RDCD = 000005
BYTE29 = 000035	BYTE80 = 000120	ERR5 = ***** GX	PR2LN = 000002 G	Q\$RDMD = 000006
BYTE3 = 000003	BYTE81 = 000121	ERR7 = ***** GX	PR2TBL = 000000RG	002 Q\$REBK = 001000
BYTE30 = 000036	BYTE82 = 000122	FIND = ***** GX	PR2X = 001226R	002 Q\$PNC = 006000
BYTE31 = 000037	BYTE83 = 000123	HANG = ***** GX	QR\$CR1 = 176420	Q\$RSC = 004000
BYTE32 = 000040	BYTE84 = 000124	HLB = ***** GX	QR\$CR2 = 176422	Q\$RSET = 000010
BYTE33 = 000041	BYTE85 = 000125	HR = ***** GX	Q\$ELBR = 176424	Q\$SM = 100000
BYTE34 = 000042	BYTE86 = 000126	INCVAL = ***** GX	Q\$ATTN = 000100	Q\$SPR = 000120
BYTE35 = 000043	BYTE87 = 000127	KILL = ***** GX	Q\$BCL = 000001	Q\$SP2 = 000340
BYTE36 = 000044	BYTE88 = 000130	LBCP = ***** GX	Q\$CCCP = 000040	RQO.EN = 000200
BYTE37 = 000045	BYTE89 = 000131	LBMRP = ***** GX	Q\$CHB = 000400	RQO.VA = 020000

CPPR- M1110 27-MAR-80 14:44 PAGE 6-3
SYMBOL TABLE

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

RTNPT. = ***** GX.	T\$DISK = 000200	WORD1 = 000002	WORD40 = 000120	WORD71 = 000216
SCAN. = ***** GX.	T\$DRD = 000004	WORD10 = 000024	WORD41 = 000122	WORD72 = 000220
SEQCS. = ***** GX.	T\$MEM = 010000	WORD11 = 000026	WORD42 = 000124	WORD73 = 000222
SEQMM. = ***** GX.	T\$FSA = 000000	WORD12 = 000030	WORD43 = 000126	WORD74 = 000224
SEQ.CI = 000010	T\$FSAB = 000004	WORD13 = 000032	WORD44 = 000130	WORD75 = 000226
S\$CLR = 000000	T\$FSAC = 000014	WORD14 = 000034	WORD45 = 000132	WORD76 = 000230
S\$LA = 000001	T\$FSB2 = 000010	WORD15 = 000036	WORD46 = 000134	WORD77 = 000232
S\$QB = 000005	T\$IB = 000026	WORD16 = 000040	WORD47 = 000136	WORD78 = 000234
S\$QR = 000006	T\$IBAR = 000024	WORD17 = 000042	WORD48 = 000140	WORD79 = 000236
S\$QX = 000004	T\$IBE = 020000	WORD18 = 000044	WORD49 = 000142	WORD8 = 000020
S\$SR = 000007	T\$IBF = 040000	WORD19 = 000046	WORD5 = 000012	WORD80 = 000240
S\$S1 = 000010	T\$ICD = 000040	WORD2 = 000004	WORD50 = 000144	WORD81 = 000242
S\$S2 = 000014	T\$MODE = 004000	WORD20 = 000050	WORD51 = 000146	WORD82 = 000244
TD\$CTR = 176370	T\$OB = 000036	WORD21 = 000052	WORD52 = 000150	WORD83 = 000246
TD\$CTW = 176360	T\$OBE = 004000	WORD22 = 000054	WORD53 = 000152	WORD84 = 000250
TD\$INL = 004000	T\$OBF = 010000	WORD23 = 000056	WORD54 = 000154	WORD85 = 000252
TD\$MEM = 000270	T\$OBRA = 000034	WORD24 = 000060	WORD55 = 000156	WORD86 = 000254
TD\$OAR = 176344	T\$OBWA = 000032	WORD25 = 000062	WORD56 = 000160	WORD87 = 000256
TD\$OTR = 176346	T\$OUTA = 100000	WORD26 = 000064	WORD57 = 000162	WORD88 = 000260
TD\$ORD = 000274	T\$RBD0 = 000200	WORD27 = 000066	WORD58 = 000164	WORD89 = 000262
TD\$SW = 176376	T\$RNB = 000040	WORD28 = 000070	WORD59 = 000166	WORD9 = 000022
TD\$TAR = 176372	T\$RSET = 040000	WORD29 = 000072	WORD6 = 000014	WORD90 = 000264
TD\$TAW = 176362	T\$SC = 000022	WORD3 = 000006	WORD60 = 000170	WORD91 = 000266
TD\$TDR = 176374	T\$SCLK = 020000	WORD30 = 000074	WORD61 = 000172	WORD92 = 000270
TD\$TDW = 176364	T\$SEG1 = 000000	WORD31 = 000076	WORD62 = 000174	WORD93 = 000272
T\$AD = 000020	T\$SEG2 = 000001	WORD32 = 000100	WORD63 = 000176	WORD94 = 000274
T\$BA = 000002	T\$SEG3 = 000002	WORD33 = 000102	WORD64 = 000200	WORD95 = 000276
T\$BD = 000010	T\$S0 = 000001	WORD34 = 000104	WORD65 = 000202	WORD96 = 000300
T\$BS0 = 100000	T\$UBUS = 100000	WORD35 = 000106	WORD66 = 000204	WORD97 = 000302
T\$BT = 000020	T\$1CLK = 000400	WORD36 = 000110	WORD67 = 000206	WORD98 = 000304
T\$BTAR = 000030	T\$BBEN = 000020	WORD37 = 000112	WORD68 = 000210	WORD99 = 000306
T\$BTD = 002000	UBD.IN = 000020	WORD38 = 000114	WORD69 = 000212	WORDVAL = 000310
T\$CD = 000100	UNPK = ***** GX.	WORD39 = 000116	WORD7 = 000016	XTREAD = 001000
T\$CLK = 002000	WORD0 = 000000	WORD4 = 000010	WORD70 = 000214	XTURTE = 000400

. ABS. 000000 000
000000 001
CPPR. 001564 002.
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3671 WORDS. (15 PAGES)
DYNAMIC MEMORY: 4916 WORDS. (18 PAGES)
ELAPSED TIME: 00:00:50
CPPR,CPPR/SP=C20,11M,C20,11CPPR

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

1
2 000000 .TITLE BCE
3 .PSECT BCE
4 .LIST MEB
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36 000000 BCTBL:
37 000000 123 124 .ASCII /ST/ ;STORE INTO A REG
38 000002 000000G .WORD ST3
39 000004 122 105 .ASCII /RE/ ;READ FROM A REG
40 000006 000000G .WORD RE3
41 000010 122 123 .ASCII /RS/ ;RESET HQR
42 000012 000000G .WORD RS3
43 000003 BCNUM: = <.-BCTBL>/4

```



```

45      ;
46      ;
47      ;       BCE ROUTINES
48      ;
49      ;
50      BCE::
51      000014 004767 000000G  JSR  PC,FIND  ;LOCATE THE COMMAND IN THE COMMAND LINE
52      000020 103003          BCC  1$          ;OK, CONTINUE
53      000022 004767 000000G  JSR  PC,ERR3
54      000026 000422          BR   BCEXX
55      000030 022700 000002    1$: CMP  #2,R0      ;COMMANDS ARE 2 CHARS
56      000034 001403          BEQ  2$
57      000036 004767 000000G  JSR  PC,ERR8
58      000042 000414          BR   BCEXX
59      ;
60      000044 012700 000003    2$: MOV  #BCNUM,R0    ;R0 = NUMBER OF COMMANDS
61      000050 012702 000000*    MOV  #BCTBL,R2    ;R2 -> TABLE OF COMMAND MNEMONICS
62      000054 004767 000000G  JSR  PC,SCAN
63      000060 103003          BCC  3$          ;FIND MATCH IN TABLE
64      000062 004767 000000G  JSR  PC,ERR12
65      000066 000402          BR   BCEXX
66      ;
67      ;
68      ;       JUMP TO THE ROUTINE THAT GOVERNS THE COMMAND
69      000070 000171 000000    3$: JMP  @ (R1)
70      ;
71      ;
72      ;       LOCAL BCE LOOP, INCLUDE PROMPT FOR BCE COMMAND
73      ;
74      ;
75      000074          BCEXX::
76      000074 012767 041502 000000G  MOV  #"BC,GCMBLK+G,DPRM+2 ;MOVE BCE NAME TO GCM,BLOCK
77      000102 004767 000000G  JSR  PC,GCONLY  ;PROMPT
78      000106 004767 000000G  JSR  PC,FIND
79      000112 103003          BCC  1$          ;LOCATE THE COMMAND IN THE COMMAND LINE
80      000114 004767 000000G  JSR  PC,ERR3
81      000120 000765          BR   BCEXX
82      000122 022700 000002    1$: CMP  #2,R0      ;COMMANDS ARE 2 CHARS
83      000126 001403          BEQ  2$
84      000130 004767 000000G  JSR  PC,ERR8
85      000134 000757          BR   BCEXX
86      ;
87      000136 012700 000003    2$: MOV  #BCNUM,R0    ;R0 = NUMBER OF COMMANDS
88      000142 012702 000000*    MOV  #BCTBL,R2    ;R2 -> TABLE OF COMMAND MNEMONICS
89      000146 004767 000000G  JSR  PC,SCAN
90      000152 103002          BCC  3$          ;FIND MATCH IN TABLE
91      000154 000167 000000G  JMP  COMXX
92      ;
93      ;
94      ;       JUMP TO THE ROUTINE THAT GOVERNS THE COMMAND
95      000160 000171 000000    3$: JMP  @ (R1)
96      ;
97      000001          .END

```

ALUCKE = 040000	BYTE39 = 000047	BYTE90 = 000132	N = 000144	Q\$QLR = 000001
ALUOE = 004000	BYTE4 = 000004	BYTE91 = 000133	PLB = 000010	Q\$QW = 000042
A01 = 010000	BYTE40 = 000050	BYTE92 = 000134	PLC = 000020	Q\$RDCD = 000005
BCE = 000014RG	002·BYTE41 = 000051	BYTE93 = 000135	PLD = 000030	Q\$RDMD = 000006
BCEXX = 000074RG	002·BYTE42 = 000052	BYTE94 = 000136	PLRW = 000200	Q\$REBK = 001000
BCNUM = 000003	BYTE43 = 000053	BYTE95 = 000137	PLREN = 000200	Q\$RNC = 006000
BCTBL = 000000R	002·BYTE44 = 000054	BYTE96 = 000140	Q\$CR1 = 176420	Q\$RSC = 004000
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	Q\$CR2 = 176422	Q\$RSET = 000010
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	Q\$CLBR = 176424	Q\$SM = 100000
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	Q\$ATTN = 000100	Q\$SP = 000120
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	Q\$BCL = 000001	Q\$SP2 = 000340
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	Q\$CCCP = 000040	RE3 = ***** GX
BIT12 = 010000	BYTE5 = 000005	CBKCLK = 000400	Q\$CHB = 000400	RGQ·EN = 000200
BIT13 = 020000	BYTE50 = 000062	CNOBRE = 100000	Q\$CHRL = 000200	RGQ·VA = 020000
BIT14 = 040000	BYTE51 = 000063	COMXX = ***** GX	Q\$CLR = 000040	RS3 = ***** GX
BIT15 = 100000	BYTE52 = 000064	CPCCEN = 010000	Q\$CNC = 030000	SCAN = ***** GX
BIT2 = 000004	BYTE53 = 000065	CPREAD = 040000	Q\$CP = 000060	SEQ·CI = 000010
BIT3 = 000010	BYTE54 = 000066	CPWRT = 020000	Q\$CPCC = 000010	ST3 = ***** GX
BIT4 = 000020	BYTE55 = 000067	CSADRD = 000004	Q\$CP2 = 000260	STCLR = 000000
BIT5 = 000040	BYTE56 = 000070	CSEQCI = 100000	Q\$CSC = 010000	SLA = 000001
BIT6 = 000100	BYTE57 = 000071	CSOE = 000040	Q\$CSEL = 000360	S\$OB = 000005
BIT7 = 000200	BYTE58 = 000072	CSWRT = 000100	Q\$CSET = 000002	S\$OR = 000006
BIT8 = 000400	BYTE59 = 000073	DB·RD = 000001	Q\$CSP = 020000	S\$OX = 000004
BIT9 = 001000	BYTE6 = 000006	DB\$CPP = 001457	Q\$DMA = 000001	S\$SR = 000007
BYTE0 = 000000	BYTE60 = 000074	DB\$SPT = 000026	Q\$ENBK = 040000	S\$S1 = 000010
BYTE1 = 000001	BYTE61 = 000075	DB\$TPC = 000023	Q\$ENOP = 020000	S\$S2 = 000014
BYTE10 = 000012	BYTE62 = 000076	DISPGS = 100000	Q\$FAL = 004000	TD\$CTR = 176370
BYTE11 = 000013	BYTE63 = 000077	DMARUR = 000005	Q\$FC = 000045	TD\$CTW = 176360
BYTE12 = 000014	BYTE64 = 000100	DMARRD = 000003	Q\$FO = 000044	TD\$INL = 004000
BYTE13 = 000015	BYTE65 = 000101	DMARUR = 000004	Q\$FP = 000046	TD\$MEM = 000270
BYTE14 = 000016	BYTE66 = 000102	ENBR = 010000	Q\$HBF = 000002	TD\$OAR = 176344
BYTE15 = 000017	BYTE67 = 000103	ERR12 = ***** GX	Q\$ICP = 000006	TD\$OTR = 176346
BYTE16 = 000020	BYTE68 = 000104	ERR3 = ***** GX	Q\$IH = 000003	TD\$QRD = 000274
BYTE17 = 000021	BYTE69 = 000105	ERR8 = ***** GX	Q\$IHRL = 000002	TD\$SW = 176376
BYTE18 = 000022	BYTE7 = 000007	FIND = ***** GX	Q\$INRP = 000007	TD\$TAG = 176372
BYTE19 = 000023	BYTE70 = 000106	GCMBLK = ***** GX	Q\$LBD = 001000	TD\$TAG = 176362
BYTE2 = 000002	BYTE71 = 000107	GCONLY = ***** GX	Q\$LBDP = 001001	TD\$TDR = 176374
BYTE20 = 000024	BYTE72 = 000110	G·DPRM = ***** GX	Q\$LBP = 000001	TD\$TDW = 176364
BYTE21 = 000025	BYTE73 = 000111	LOC·EN = 000100	Q\$LDCD = 000003	T\$AD = 000020
BYTE22 = 000026	BYTE74 = 000112	LOC·UA = 040000	Q\$LDMD = 000004	T\$BA = 000002
BYTE23 = 000027	BYTE75 = 000113	LOC·UB = 100000	Q\$LDPP = 002000	T\$BD = 000010
BYTE24 = 000030	BYTE76 = 000114	MAREN1 = 000001	Q\$LHP = 010000	T\$BSO = 100000
BYTE25 = 000031	BYTE77 = 000115	MAREN2 = 004000	Q\$MNC = 140000	T\$BT = 000020
BYTE26 = 000032	BYTE78 = 000116	MARLOD = 100000	Q\$MR = 000052	T\$BTAR = 000030
BYTE27 = 000033	BYTE79 = 000117	MAROUT = 000002	Q\$MRP = 000040	T\$BTD = 002000
BYTE28 = 000034	BYTE8 = 000010	MAR·LO = 002000	Q\$MRP2 = 000240	T\$CD = 000100
BYTE29 = 000035	BYTE80 = 000120	MAR·OU = 000040	Q\$MSC = 040000	T\$CLK = 002000
BYTE3 = 000003	BYTE81 = 000121	MBKALL = 001000	Q\$MSET = 000004	T\$DISK = 000200
BYTE30 = 000036	BYTE82 = 000122	MBKCLK = 000400	Q\$MSP = 100000	T\$DRD = 000004
BYTE31 = 000037	BYTE83 = 000123	MMADRD = 000100	Q\$NCLK = 176000	T\$EMEM = 010000
BYTE32 = 000040	BYTE84 = 000124	MMLEFT = 000002	Q\$PP = 000100	T\$FSAA = 000000
BYTE33 = 000041	BYTE85 = 000125	MMDE = 000004	Q\$PPSW = 000320	T\$FSAB = 000004
BYTE34 = 000042	BYTE86 = 000126	MMWRT = 000010	Q\$PP2 = 000300	T\$FSAC = 000014
BYTE35 = 000043	BYTE87 = 000127	MNOBRE = 100000	Q\$QHLT = 000013	T\$FSB2 = 000010
BYTE36 = 000044	BYTE88 = 000130	MFEN1 = 000001	Q\$QL = 000043	T\$IB = 000026
BYTE37 = 000045	BYTE89 = 000131	MREN2 = 020000	Q\$QLA = 000053	T\$IBAR = 000024
BYTE38 = 000046	BYTE9 = 000011	MEYN = 000040	Q\$QLB = 000054	T\$IBE = 020000

BCE.....MACRO: M1110 27-MAR-80 14:35 PAGE: 6-2.
SYMBOL TABLE

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

T\$IBF = 040000	WORD11 = 000026	WORD34 = 000104	WORD57 = 000162	WORD8 = 000020
T\$ICD = 000040	WORD12 = 000030	WORD35 = 000106	WORD58 = 000164	WORD80 = 000240
T\$MODE = 004000	WORD13 = 000032	WORD36 = 000110	WORD59 = 000166	WORD81 = 000242
T\$OB = 000036	WORD14 = 000034	WORD37 = 000112	WORD6 = 000014	WORD82 = 000244
T\$OBE = 004000	WORD15 = 000036	WORD38 = 000114	WORD60 = 000170	WORD83 = 000246
T\$OBF = 010000	WORD16 = 000040	WORD39 = 000116	WORD61 = 000172	WORD84 = 000250
T\$OBRA = 000034	WORD17 = 000042	WORD4 = 000010	WORD62 = 000174	WORD85 = 000252
T\$OBWA = 000032	WORD18 = 000044	WORD40 = 000120	WORD63 = 000176	WORD86 = 000254
T\$OUTA = 100000	WORD19 = 000046	WORD41 = 000122	WORD64 = 000200	WORD87 = 000256
T\$RBD0 = 000200	WORD2 = 000004	WORD42 = 000124	WORD65 = 000202	WORD88 = 000260
T\$RNB = 000040	WORD20 = 000050	WORD43 = 000126	WORD66 = 000204	WORD89 = 000262
T\$RSET = 040000	WORD21 = 000052	WORD44 = 000130	WORD67 = 000206	WORD9 = 000022
T\$SC = 000022	WORD22 = 000054	WORD45 = 000132	WORD68 = 000210	WORD90 = 000264
T\$SCLK = 020000	WORD23 = 000056	WORD46 = 000134	WORD69 = 000212	WORD91 = 000266
T\$SEG1 = 000000	WORD24 = 000060	WORD47 = 000136	WORD7 = 000016	WORD92 = 000270
T\$SEG2 = 000001	WORD25 = 000062	WORD48 = 000140	WORD70 = 000214	WORD93 = 000272
T\$SEG3 = 000002	WORD26 = 000064	WORD49 = 000142	WORD71 = 000216	WORD94 = 000274
T\$SO = 000001	WORD27 = 000066	WORD5 = 000012	WORD72 = 000220	WORD95 = 000276
T\$UBUS = 100000	WORD28 = 000070	WORD50 = 000144	WORD73 = 000222	WORD96 = 000300
T\$1CLK = 000400	WORD29 = 000072	WORD51 = 000146	WORD74 = 000224	WORD97 = 000302
T\$BBEN = 000020	WORD3 = 000006	WORD52 = 000150	WORD75 = 000226	WORD98 = 000304
UBD.IN = 000020	WORD30 = 000074	WORD53 = 000152	WORD76 = 000230	WORD99 = 000306
WORD0 = 000000	WORD31 = 000076	WORD54 = 000154	WORD77 = 000232	WRDVAL = 000310
WORD1 = 000002	WORD32 = 000100	WORD55 = 000156	WORD78 = 000234	XTREAD = 001000
WORD10 = 000024	WORD33 = 000102	WORD56 = 000160	WORD79 = 000236	XTURTE = 000400

. ABS. 000000 000
000000 001
BCE: 000164 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3103 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:42
BCE,BEE/SP=C20.1JIM.C20.1JBCE

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

1      .TITLE--BCREST...
2 000000 .PSECT--BCREST
3      .LIST--MEB
4
5      :
6      :
7      :
8      :
9      :
10     :
11     :
12     :
13     :
14     :
15     :
16     :
17     :
18     :
19 000000 .STORE
20 000000      103      061      ST3TBL::
21 000002      000150'      .ASCII /C1/      :CONTROL AND STATUS REG #1
22 000004      103      062      .WORD ST3C1
23 000006      000150'      .ASCII /C2/      :CONTROL AND STATUS REG #2
24 000010      114      102      .WORD ST3C2
25 000012      000170'      .ASCII /LB/      :LOD BUS REG
26 000003      000003      .WORD ST3LB
27      ST3LN == <.-ST3TBL>/4
28      :
29      :
30      :
31      :
32 000014 .READ
33 000014      103      061      TABLE OF VALID REGISTER MNEMONICS AND ASSOCIATED ROUTINE
34 000016      000310'      ADDRESSES
35 000020      103      062      RE3TBL::
36 000022      000316'      .ASCII /C1/      :CONTROL AND STATUS REG #1
37 000024      114      102      .WORD RE3C1
38 000026      000324'      .ASCII /C2/      :CONTROL AND STATUS REG #2
39 000003      000003      .WORD RE3C2
40      RE3LN == <.-RE3TBL>/4
41      :
42      :
43      :
44      :
45      :
46      :
47      :
48      :
49 000030 .STORE
50 000030      004767 000000G .PERFORM THIRD LEVEL PARSING
51 000034      103004      EG. IN THE COMMAND
52 000036      004767 000000G .BC>ST 0 C1
53 000042      000167 000145 .PARSE THE '0'
54 000046      004767 000000G .
55 000052      103004      ST3::
56 000054      004767 000000G .JSR PC.FIND      :FIND A NON-BLANK IN COMMAND LINE
57 000060      000167 000130 .BCC 1$      :OK, CONTINUE
      .JSR PC.ERR4      :NOTHING THERE
      .JMP ST3X      :RETURN TO TOP OF LOOP (PROMPT)
      1$: .JSR PC.PACK      :CONVERT VALUE IN COMMAND LINE TO BINARY
      .BCC 2$
      .JSR PC.ERR5
      .JMP ST3X

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

58      ;
59      ;
60      ;
61      ;
62      000064 004767 000000G 2$: JSR PC,FIND ;LOCATE A NON-BLANK
63      000070 103004 BCC 3$
64      000072 004767 000000G JSR PC,ERR4
65      000076 000167 000112 JMP ST3X
66      ;
67      ;
68      ;
69      ;
70      000102 012700 000003 3$: MOV #ST3LN,R0 ;NUMBER OF TABLE ENTRIES
71      000106 012700 000000 MOV #ST3BL,R2 ;R2 -> TABLE
72      000112 004767 000000G JSR PC,SCAN ;MATCH AGAINST COMMAND LINE
73      000116 103004 BCC 4$ ;OK, CONTINUE
74      000120 004767 000000G JSR PC,ERR6
75      000124 000167 000064 JMP ST3X
76      ;
77      ;
78      ;
79      ;
80      ;
81      ;
82      ;
83      ;
84      000130 010167 000000G 4$: MOV R1,RTNPT ;SAVE POINTER TO RTN
85      000134 004767 000000G JSR PC,LOOPR ;LOOP?
86      000140 016701 000000G ST3IN: MOV RTNPT,R1 ;POINT TO ROUTINE
87      000144 000171 000000 JMP @R1 ;EXECUTE ROUTINE
88      ;
89      ;
90      ;
91      ;
92      000150 ST3C1: CONTROL AND STATUS REG #1
93      000150 016767 000000G 176420 MOV BINWD,QR#CR1 ;MOVE USER WORD TO CSR #1
94      000156 000410 BR ST3LP ;TEST FOR REPEAT
95      ;
96      ;
97      ;
98      000160 016767 000000G 176422 ST3C2: MOV BINWD,QR#CR2 ;MOVE USER WORD TO CSR #2
99      000166 000404 BR ST3LP ;TEST FOR REPEAT
100      ;
101      ;
102      ;
103      000170 016767 000000G 176424 ST3LB: MOV BINWD,QR#LBR ;MOVE USER WORD TO LOD BUS REG
104      000176 000400 BR ST3LP ;TEST FOR REPEAT
105      ;
106      ;
107      000200 ST3LP:
108      000200 032767 000000G 000000G BIT #LOOP,BASE ;REPEAT?
109      000206 001402 BEQ ST3X ;NO
110      000210 000167 177724 JMP ST3IN ;AND REPEAT
111      ;
112      000214 ST3X:
113      000214 004767 000000G JSR PC,KILL ;KILL AST
114      000220 000167 000000G JMP BCEXX

```

Approved For Release 2005/07/12 : CIA-RDP85-00

```

116      ;
117      ;
118      ;
119      ; READ
120      ; PERFORM THIRD-LEVEL PARSING
121      ; EG. IN THE COMMAND
122      ; BC>RE C1
123      ; PARSE THE 'C1'
124      ;
125      ;
126      RE3:: JSR PC,FIND      ;FIND A REG. MNEMONIC IN COMMAND LINE
127      BCC 1$      ;OK, CONTINUE
128      JSR PC,ERR4      ;NOTHING THERE
129      JMP RE3X      ;RETURN TO TOP OF LOOP (PROMPT)
130      ;
131      ;
132      ; MATCH THE REGISTER MNEMONIC FROM THE COMMAND LINE AGAINST
133      ; THE TABLE OF VALID ADDRESSES.
134      1$: MOV #RE3LN,R0      ;NUMBER OF TABLE ENTRIES
135      MOV #RE3TBL,R2      ;R2 -> TABLE
136      JSR PC,SCAN      ;MATCH AGAINST COMMAND LINE
137      BCC 2$      ;OK, CONTINUE
138      JSR PC,ERR6
139      JMP RE3X
140      ;
141      ;
142      ; SAVE THE POINTER TO THE ROUTINE ASSOCIATED WITH THE
143      ; REGISTER. R1 -> ROUTINE ADDRESS.
144      ; CALL ROUTINE TO SCAN COMMAND LINE FOR LOOP INDICATOR.
145      ; EG. BC>RE C1 L
146      ; LOOP FLAG WILL BE SET IF INDICATOR IS PRESENT
147      ; JUMP TO ROUTINE TO LOAD REGISTER.
148      2$: MOV R1,RTNPT      ;SAVE POINTER TO RTN
149      JSR PC,LOOPR      ;LOOP?
150      RE3IN: MOV RTNPT,R1      ;POINT TO ROUTINE
151      JMP @R1      ;EXECUTE ROUTINE
152      ;
153      ;
154      ;
155      ; CONTROL AND STATUS REG #1
156      RE3C1:: MOV QR#CR1,R1      ;GET CSR #1
157      BR RE3PUT      ;AND PRINT
158      ;
159      ;
160      ; CONTROL AND STATUS REG #2
161      RE3C2:: MOV QR#CR2,R1      ;GET CSR #2
162      BR RE3PUT      ;AND PRINT
163      ;
164      ;
165      ;
166      ; LOD BUS REG
167      RE3LB:: MOV QR#LBR,R1      ;GET LOD BUS REG
168      BR RE3PUT      ;AND PRINT
169      ;
170      ;
171      ;
172      ;

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

173 000332.          RE3PUT:
174 000332. 032767 000000G-000000G. BIT. #ONCE,BASE. ;PRINTED ONCE?
175 000340 001011 BNE. 1$ ;YES, SKIP
176 000342. 052767 000000G-000000G. BIS. #ONCE,BASE. ;SET FLAG FOR PRINTED ONCE
177 000350 012705 000000G. MOV. *PRINT,R5 ;POINT TO PRINT LINE
178 000354 004767 000000G. JSR. PC,UNPK. ;CONVERT VALUE IN R1 FOR PRINTING
179 000360 004767 000000G. JSR. PC,CONSOL. ;PRINT ON CONSOLE
180
181 000364 032767 000000G-000000G. 1$: BIT. #LOOP,BASE. ;REPEAT?
182 000372. 001402. BEQ. RE3X ;NO
183 000374 000167 177700 JMP. RE3IN. ;AND REPEAT
184
185 000400          RE3X:
186 000400 042767 000000G-000000G. BIC. #ONCE,BASE. ;CLEAR PRINT CONTROL FLAG
187 000406 004767 000000G. JSR. PC,KILL. ;KILL AST
188 000412. 000167 000000G. JMP. BCEXX
189
190
191
192
193
194          RESET,HQR.
195          RS3::
196 000416 005046          CLR. -(SP) ;CLEAR NO BITS IN CSR1 FW
197 000420 012746 000010 MOV. #Q$RSET,-(SP) ;SET RESET FW
198 000424 004767 000000G. JSR. PC,CSR1 ;CHANGE CSR1 FW
199 000430 012746 000010 MOV. #Q$RSET,-(SP) ;CLEAR RESET FW
200 000434 005046          CLR. -(SP) ;DON'T SET ANYTHING FW
201 000436 004767 000000G. JSR. PC,CSR1 ;CHANGE CSR1 FW
202 000442. 000167 000000G. JMP. BCEXX
203
203          .END.

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

ALUCKE = 000000	BYTE4 = 000004	BYTE91 = 000133	N = 000144	Q\$QL = 000043
ALUOE = 000000	BYTE40 = 000050	BYTE92 = 000134	ONCE = ***** GX	Q\$OLA = 000053
A01 = 010000	BYTE41 = 000051	BYTE93 = 000135	PACK = ***** GX	Q\$QLR = 000001
BASE = ***** GX	BYTE42 = 000052	BYTE94 = 000136	PLB = 000010	Q\$QLB = 000054
BCEXX = ***** GX	BYTE43 = 000053	BYTE95 = 000137	PLC = 000020	Q\$QLW = 000042
BINWD = ***** GX	BYTE44 = 000054	BYTE96 = 000140	PLD = 000030	Q\$RDCD = 000005
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	PLRWR = 000200	Q\$RDM = 000006
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	PLREN = 000200	Q\$REBK = 001000
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	PRINT = ***** GX	Q\$RNC = 006000
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	QR\$CR1 = 176420	Q\$RSC = 004000
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	QR\$CR2 = 176422	Q\$RSET = 000010
BIT12 = 010000	BYTE5 = 000005	CBKCLK = 000400	QR\$LBR = 176424	Q\$SM = 100000
BIT13 = 020000	BYTE50 = 000062	CNOBRE = 100000	Q\$ATTN = 000100	Q\$SP = 000120
BIT14 = 040000	BYTE51 = 000063	CONSOL = ***** GX	Q\$BCL = 000001	Q\$SP2 = 000340
BIT15 = 100000	BYTE52 = 000064	CPCEN = 010000	Q\$CCCP = 000040	RE3 = 000224RG 002
BIT2 = 000004	BYTE53 = 000065	CPREAD = 040000	Q\$CHB = 000400	RE3C1 = 000310RG 002
BIT3 = 000010	BYTE54 = 000066	CPWRTE = 020000	Q\$CHRL = 000200	RE3C2 = 000316RG 002
BIT4 = 000020	BYTE55 = 000067	CSADRD = 000004	Q\$CLR = 000040	RE3IN = 000300R 002
BIT5 = 000040	BYTE56 = 000070	CSEDCI = 100000	Q\$CNC = 030000	RE3LB = 000324RG 002
BIT6 = 000100	BYTE57 = 000071	CSOE = 000040	Q\$CP = 000060	RE3LN = 000003 G
BIT7 = 000200	BYTE58 = 000072	CSR1 = ***** GX	Q\$CPC = 000010	RE3PUT = 000332R 002
BIT8 = 000400	BYTE59 = 000073	CSWRTE = 000100	Q\$CP2 = 000260	RE3TBL = 000014RG 002
BIT9 = 001000	BYTE6 = 000006	DBR RD = 000001	Q\$CSC = 010000	RE3X = 000400R 002
BYTE0 = 000000	BYTE60 = 000074	DB\$CPP = 001457	Q\$CSEL = 000360	RQO EM = 000200
BYTE1 = 000001	BYTE61 = 000075	DB\$SPT = 000026	Q\$CSET = 000002	RQO VA = 020000
BYTE10 = 000012	BYTE62 = 000076	DB\$TPC = 000023	Q\$DMA = 000001	R33 = 000416RG 002
BYTE11 = 000013	BYTE63 = 000077	DISPGS = 100000	Q\$ENBK = 040000	RTNPT = ***** GX
BYTE12 = 000014	BYTE64 = 000100	DMAWRD = 000005	Q\$ENOP = 020000	SEQ CI = 000010
BYTE13 = 000015	BYTE65 = 000101	DMARRD = 000004	Q\$FAL = 004000	ST3 = 000030RG 002
BYTE14 = 000016	BYTE66 = 000102	ENBR = 010000	Q\$FC = 000045	ST3C1 = 000150RG 002
BYTE15 = 000017	BYTE67 = 000103	ERR4 = ***** GX	Q\$FO = 000044	ST3C2 = 000160RG 002
BYTE16 = 000020	BYTE68 = 000104	ERR5 = ***** GX	Q\$FP = 000046	ST3IN = 000140R 002
BYTE17 = 000021	BYTE69 = 000105	ERR6 = ***** GX	Q\$HBF = 000002	ST3LB = 000170RG 002
BYTE18 = 000022	BYTE7 = 000007	FIND = ***** GX	Q\$ICP = 000006	ST3LN = 000003 G
BYTE19 = 000023	BYTE70 = 000106	KILL = ***** GX	Q\$IHB = 000003	ST3LP = 000200R 002
BYTE2 = 000002	BYTE71 = 000107	LOC EN = 000100	Q\$IHL = 000002	ST3TBL = 000000RG 002
BYTE20 = 000024	BYTE72 = 000110	LOC WA = 040000	Q\$IMRP = 000007	ST3X = 000214R 002
BYTE21 = 000025	BYTE73 = 000111	LOC WB = 100000	Q\$LBD = 001000	S\$CLR = 000000
BYTE22 = 000026	BYTE74 = 000112	LOOP = ***** GX	Q\$LBDP = 001001	S\$LA = 000001
BYTE23 = 000027	BYTE75 = 000113	LOOPR = ***** GX	Q\$LBP = 000001	S\$QB = 000005
BYTE24 = 000030	BYTE76 = 000114	MAREN1 = 000001	Q\$LCD = 000003	S\$QR = 000006
BYTE25 = 000031	BYTE77 = 000115	MAREN2 = 000004	Q\$LDMD = 000004	S\$OX = 000004
BYTE26 = 000032	BYTE78 = 000116	MARLOD = 010000	Q\$LDPP = 002000	S\$SR = 000007
BYTE27 = 000033	BYTE79 = 000117	MAROUT = 000002	Q\$LHP = 010000	S\$S1 = 000010
BYTE28 = 000034	BYTE8 = 000010	MAR LO = 002000	Q\$MNC = 140000	S\$S2 = 000014
BYTE29 = 000035	BYTE80 = 000120	MAR OU = 000040	Q\$MR = 000052	TD\$CTR = 176370
BYTE3 = 000003	BYTE81 = 000121	MBKALL = 001000	Q\$MRP = 000040	TD\$CTW = 176360
BYTE30 = 000036	BYTE82 = 000122	MBKCLK = 000400	Q\$MRP2 = 000240	TD\$INL = 004000
BYTE31 = 000037	BYTE83 = 000123	MMAORD = 000100	Q\$MSC = 040000	TD\$MEM = 000270
BYTE32 = 000040	BYTE84 = 000124	MMLFT = 000002	Q\$MSET = 000004	TD\$MAR = 176344
BYTE33 = 000041	BYTE85 = 000125	MMOE = 000004	Q\$MSP = 100000	TD\$OTR = 176346
BYTE34 = 000042	BYTE86 = 000126	MMLWRTE = 000010	Q\$NCLK = 176000	TD\$ORD = 000274
BYTE35 = 000043	BYTE87 = 000127	MNOBRE = 100000	Q\$PP = 000100	TD\$SL = 176370
BYTE36 = 000044	BYTE88 = 000130	MREN1 = 000001	Q\$PPSW = 000320	TD\$STAR = 176372
BYTE37 = 000045	BYTE89 = 000131	MREN2 = 020000	Q\$PP2 = 000300	TD\$TAU = 176362
BYTE38 = 000046	BYTE9 = 000011	MSYN = 000040	Q\$OHLT = 000013	TD\$TDR = 176374
BYTE39 = 000047	BYTE90 = 000132			

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

BCREST: M1110 27-MAR-80 14:35 PAGE 6-3
SYMBOL: TA

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

TD\$TDW= 176364	T\$RNB= 000040	WORD23= 000056	WORD5 = 000012	WORD76= 000230
T\$AD= 000020	T\$RSET= 040000	WORD24= 000060	WORD50= 000144	WORD77= 000232
T\$BA= 000002	T\$SC= 000022	WORD25= 000062	WORD51= 000146	WORD78= 000234
T\$BD= 000010	T\$SCLK= 020000	WORD26= 000064	WORD52= 000150	WORD79= 000236
T\$BSO= 100000	T\$SEG1= 000000	WORD27= 000066	WORD53= 000152	WORD8 = 000020
T\$BT= 000020	T\$SEG2= 000001	WORD28= 000070	WORD54= 000154	WORD80= 000240
T\$BTAR= 000030	T\$SEG3= 000002	WORD29= 000072	WORD55= 000156	WORD81= 000242
T\$BTD= 002000	T\$SO= 000001	WORD3 = 000005	WORD56= 000160	WORD82= 000244
T\$CD= 000100	T\$UBUS= 100000	WORD30= 000074	WORD57= 000162	WORD83= 000246
T\$CLK= 002000	T\$1CLK= 000400	WORD31= 000076	WORD58= 000164	WORD84= 000250
T\$DISK= 000200	T\$BBEN= 000020	WORD32= 000100	WORD59= 000166	WORD85= 000252
T\$DRD= 000004	UBD, IN= 000020	WORD33= 000102	WORD6 = 000014	WORD86= 000254
T\$EMEM= 010000	UNPK= ***** GX	WORD34= 000104	WORD60= 000170	WORD87= 000256
T\$FSAB= 000000	WORD0 = 000000	WORD35= 000106	WORD61= 000172	WORD88= 000260
T\$FSAB= 000004	WORD1 = 000002	WORD36= 000110	WORD62= 000174	WORD89= 000262
T\$FSAC= 000014	WORD10= 000024	WORD37= 000112	WORD63= 000176	WORD9 = 000022
T\$FSB2= 000010	WORD11= 000026	WORD38= 000114	WORD64= 000200	WORD90= 000264
T\$IB= 000026	WORD12= 000030	WORD39= 000116	WORD65= 000202	WORD91= 000266
T\$IBAR= 000024	WORD13= 000032	WORD4 = 000010	WORD66= 000204	WORD92= 000270
T\$IBE= 020000	WORD14= 000034	WORD40= 000120	WORD67= 000206	WORD93= 000272
T\$IBF= 040000	WORD15= 000036	WORD41= 000122	WORD68= 000210	WORD94= 000274
T\$ICD= 000040	WORD16= 000040	WORD42= 000124	WORD69= 000212	WORD95= 000276
T\$MODE= 004000	WORD17= 000042	WORD43= 000126	WORD7 = 000016	WORD96= 000300
T\$OB= 000036	WORD18= 000044	WORD44= 000130	WORD70= 000214	WORD97= 000302
T\$OBE= 004000	WORD19= 000046	WORD45= 000132	WORD71= 000216	WORD98= 000304
T\$OBF= 010000	WORD2 = 000004	WORD46= 000134	WORD72= 000220	WORD99= 000306
T\$OBRA= 000034	WORD20= 000050	WORD47= 000136	WORD73= 000222	WORDVAL= 000310
T\$OBWA= 000032	WORD21= 000052	WORD48= 000140	WORD74= 000224	XTREAD= 001000
T\$OUTA= 100000	WORD22= 000054	WORD49= 000142	WORD75= 000226	XTWRITE= 000400
T\$RBD0= 000200				

. ABS. 000000 000
000000 001
BCREST 000446 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3196 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:42
BCREST, BCREST/SP=[20,1]IM,[20,1]BCREST

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
1
2
3 000000 .TITLE PPS
4 .PSECT PPS
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
```

HARDWARE QUERY RESOLVER: 'MANUAL' DEBUGGING AIDS
PIPELINE PROCESSORS.

PARSE THE COMMAND LINE AND TRANSFER CONTROL TO ONE OF THE
PPS SUB-MODULES OR BACK TO QMAIN. CONTROL COULD HAVE BEEN
PASSED TO PPS IN ONE OF THREE WAYS:

1. FROM QMAIN IF PPS WAS SELECTED AS THE FIRST PROCESSOR
UPON ENTRY TO THE PROGRAM.
PP LD QW 0
2. FROM QMAIN IF PPS WAS SELECTED FROM ONE OF QMAIN'S
OTHER SUB-MODULES.
CP PP LD QW 0
3. ON RETURN FROM ONE OF PPS'S SUB-MODULES
PP LD QW 0

PPS PARSES AT THE SECOND LEVEL OF CONTROL (SEE NOTES AT
QMAIN ON LEVELS OF CONTROL). IN THE EXAMPLES ABOVE, PPS
WOULD PARSE 'LD' AND TRANSFER CONTROL TO THE PPS SUB-
MODULE PPLD. IF PPS SHOULD ENCOUNTER A STRING WHICH IS
NOT A VALID COMMAND MNEMONIC, PPS CONSIDERS THE STRING TO
BE A PROCESSOR MNEMONIC AND RETURNS CONTROL TO QMAIN. EG:
PP CP LD CS 0
THE STRING 'CP' IS NOT A VALID PPS (SECOND LEVEL) COMMAND.
PPS RETURNS CONTROL TO QMAIN WHICH IN TURN WILL TRANSFER
CONTROL TO ITS SUB-MODULE CP.

PPS SUB-MODULES:
PPLD: LOAD MEMORIES.
PPPR: PRINT MEMORY CONTENTS.
PPREST: ALL OTHER PPS COMMANDS.

.MCALL WTSE\$S,CLEF\$S

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

40      ;
41      ;
42      ;
43      ;
44      ;
45      ;
46 000000      ;
47 000000      ;
48 000002 123 124      ;
49 000004 122 105      ;
50 000006 000000G      ;
51 000010 114 104      ;
52 000012 000000G      ;
53 000014 120 122      ;
54 000016 000000G      ;
55 000020 105 122      ;
56 000022 000000G      ;
57 000024 120 123      ;
58 000026 000000G      ;
59 000030 103 114      ;
60 000032 000000G      ;
61 000007      ;
62      ;
63      ;
64      ;
65      ;
66 000034 000000      ;

```

TABLE OF PPS COMMAND MNEMONICS AND THEIR ASSOCIATED ROUTINE ADDRESSES

PPTBL:

.ASCII	/ST/		:STORE INTO A REG.
.WORD	ST4		
.ASCII	/RE/		:READ FROM A REG.
.WORD	RE4		
.ASCII	/LD/		:LOAD MEMORY.
.WORD	LD4		
.ASCII	/PR/		:PRINT FROM MEMORY.
.WORD	PPPR4		
.ASCII	/ER/		:MEMORY ERASE.
.WORD	ER4		
.ASCII	/PS/		:PAGE SWITCH.
.WORD	PS4		
.ASCII	/CL/		:CALL OR LOADER.
.WORD	CL4		

PPNUM = <.-PPTBL>/4

CODE FOR MEMORY SELECTION

CODE: .WORD 0

```
68 ;
69 ;
70 ; PPS ROUTINES
71 ;
72 ;
73 000036 PPS::
74 000036 CALL FIND ;LOCATE THE COMMAND IN THE COMMAND LINE
75 000042 103003 BCC 1$ ;OK, CONTINUE
76 000044 CALL ERR3
77 000050 000422 BR PPSXX
78 000052 022700 000002 1$: CMP #2,R0 ;COMMANDS ARE 2 CHARS
79 000056 001403 BEQ 2$
80 000060 CALL ERR8
81 000064 000414 BR PPSXX ;TRY AGAIN
82 ;
83 000066 012700 000007 2$: MOV #PPNUM,R0 ;R0 = NUMBER OF COMMANDS
84 000072 012702 000000 MOV #PPTBL,R2 ;R2 -> TABLE OF COMMAND MNEMONICS
85 000076 CALL SCAN ;FIND MATCH IN TABLE
86 000102 103003 BCC 3$ ;OK, CONTINUE
87 000104 CALL ERR12 ;COMMAND NOT IN TABLE
88 000110 000402 BR PPSXX ;TRY AGAIN
89 ;
90 ; JUMP TO THE ROUTINE THAT GOVERNS THE COMMAND
91 ;
92 000112 000171 000000 3$: JMP @ (R1)
93 ;
94 ;
95 ; LOCAL PPS LOOP, INCLUDE PROMPT FOR PPS COMMAND
96 ;
97 ;
98 000116 PPSXX::
99 000116 012767 050120 000000C MOV #PP.GCMBLK+G.DPRM+2 ;MOVE PPS NAME TO GCMBLOCK
100 000124 CALL GCONLY ;PROMPT
101 000130 CALL FIND ;LOCATE THE COMMAND IN THE COMMAND LINE
102 000134 103003 BCC 1$ ;OK, CONTINUE
103 000136 CALL ERR3
104 000142 000765 BR PPSXX
105 000144 022700 000002 1$: CMP #2,R0 ;COMMANDS ARE 2 CHARS
106 000150 001403 BEQ 2$
107 000152 CALL ERR8
108 000156 000757 BR PPSXX ;TRY AGAIN
109 ;
110 000160 012700 000007 2$: MOV #PPNUM,R0 ;R0 = NUMBER OF COMMANDS
111 000164 012702 000000C MOV #PPTBL,R2 ;R2 -> TABLE OF COMMAND MNEMONICS
112 000170 CALL SCAN ;FIND MATCH IN TABLE
113 000174 103002 BCC 3$ ;OK, CONTINUE
114 000176 000167 000000G JMP COMXX ;RETURN TO MAIN, LOOK FOR PROCESSOR MNEMONIC
115 ;
116 ; JUMP TO THE ROUTINE THAT GOVERNS THE COMMAND
117 ;
118 000202 000171 000000 3$: JMP @ (R1)
```

Approved For Release 2005/07/10 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

PPS.....MACRO-M:110 27-MAR-80 15:26 PAGE 10

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

203
204

000001

.END.

ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	MREN2 = 020000	Q\$MSET = 000004
ALUOE = 004000	BYTE43 = 000053	BYTE95 = 000137	MRPCR = ***** GX	Q\$MSP = 100000
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	MSYN = 000040	Q\$NCLK = 176000
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	N = 000144	Q\$PP = 000100
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	PLB = 000010	Q\$PPSW = 000320
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	PLC = 000020	Q\$PP2 = 000300
BIT10 = 002000	BYTE48 = 000058	BYTVAL = 000144	PLD = 000030	Q\$QHLT = 000013
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	PLRWR = 000200	Q\$QL = 000043
BIT12 = 010000	BYTE5 = 000005	CBKCLK = 000400	PLR.EN = 000200	Q\$QLA = 000053
BIT13 = 020000	BYTE50 = 000062	CL4 = ***** GX	PPNUM = 000007	Q\$QLB = 000054
BIT14 = 040000	BYTE51 = 000063	CNOBRE = 100000	PPR4 = ***** GX	Q\$QLR = 000001
BIT15 = 100000	BYTE52 = 000064	CODE = 000034RG	PPSXX = 000036RG	Q\$QW = 000042
BIT2 = 000004	BYTE53 = 000065	COMXX = ***** GX	PPSXX = 000116RG	Q\$RDCD = 000005
BIT3 = 000010	BYTE54 = 000066	CPCCEN = 010000	PPTBL = 000000R	Q\$RDMD = 000006
BIT4 = 000020	BYTE55 = 000067	CPREAD = 040000	PS4 = ***** GX	Q\$REBK = 001000
BIT5 = 000040	BYTE56 = 000070	CPWRT = 020000	QREG = 000206RG	Q\$RNC = 006000
BIT6 = 000100	BYTE57 = 000071	CSADRD = 000004	Q\$CR1 = 176420	Q\$RSC = 004000
BIT7 = 000200	BYTE58 = 000072	CSECCI = 100000	Q\$CR2 = 176422	Q\$RSET = 000010
BIT8 = 000400	BYTE59 = 000073	CSDOE = 000040	Q\$CLBR = 176424	Q\$SM = 100000
BIT9 = 001000	BYTE6 = 000006	CSR1 = ***** GX	Q\$ATTN = 000100	Q\$SP = 000120
BYTE0 = 000000	BYTE60 = 000074	CSWRT = 000100	Q\$BCL = 000001	Q\$SP2 = 000340
BYTE1 = 000001	BYTE61 = 000075	DBR.RD = 000001	Q\$CCCP = 000040	RE4 = ***** GX
BYTE10 = 000012	BYTE62 = 000076	DB\$CPP = 001457	Q\$CHB = 000400	RGQ.EN = 000200
BYTE11 = 000013	BYTE63 = 000077	DB\$SPT = 000026	Q\$CHRL = 000200	RGQ.VA = 020000
BYTE12 = 000014	BYTE64 = 000100	DB\$TPC = 000023	Q\$CLR = 000040	SCAN = ***** GX
BYTE13 = 000015	BYTE65 = 000101	DISPGS = 100000	Q\$CNC = 030000	SEQ.CI = 000010
BYTE14 = 000016	BYTE66 = 000102	DMAWRD = 000005	Q\$CP = 000060	STGP = 000344RG
BYTE15 = 000017	BYTE67 = 000103	DMARRD = 000003	Q\$CPCC = 000010	ST4 = ***** GX
BYTE16 = 000020	BYTE68 = 000104	DMARWR = 000004	Q\$CP2 = 000260	S\$CLR = 000000
BYTE17 = 000021	BYTE69 = 000105	ENBR = 010000	Q\$CSC = 010000	S\$LA = 000001
BYTE18 = 000022	BYTE7 = 000007	ERR12 = ***** GX	Q\$CSEL = 000360	S\$QB = 000005
BYTE19 = 000023	BYTE70 = 000106	ERR3 = ***** GX	Q\$CSET = 000002	S\$QR = 000006
BYTE2 = 000002	BYTE71 = 000107	ERR8 = ***** GX	Q\$CSP = 020000	S\$QX = 000004
BYTE20 = 000024	BYTE72 = 000110	ER4 = ***** GX	Q\$DMA = 000001	S\$SR = 000007
BYTE21 = 000025	BYTE73 = 000111	FIND = ***** GX	Q\$ENBK = 040000	S\$S1 = 000010
BYTE22 = 000026	BYTE74 = 000112	GCMBLK = ***** GX	Q\$ENOP = 020000	S\$S2 = 000014
BYTE23 = 000027	BYTE75 = 000113	GCONLY = ***** GX	Q\$FAL = 004000	TD\$CTR = 176370
BYTE24 = 000030	BYTE76 = 000114	G.DPRM = ***** GX	Q\$FC = 000045	TD\$CTW = 176360
BYTE25 = 000031	BYTE77 = 000115	LBMRP = ***** GX	Q\$FO = 000044	TD\$INL = 004000
BYTE26 = 000032	BYTE78 = 000116	LD4 = ***** GX	Q\$FP = 000046	TD\$MEM = 000270
BYTE27 = 000033	BYTE79 = 000117	LOC.EN = 000100	Q\$HBF = 000002	TD\$OAR = 176344
BYTE28 = 000034	BYTE8 = 000010	LOC.WA = 040000	Q\$ICP = 000006	TD\$OTR = 176346
BYTE29 = 000035	BYTE80 = 000120	LOC.WB = 100000	Q\$IHB = 000003	TD\$ORD = 000274
BYTE3 = 000003	BYTE81 = 000121	MAREN1 = 000001	Q\$IHRL = 000002	TD\$SW = 176376
BYTE30 = 000036	BYTE82 = 000122	MAREN2 = 004000	Q\$IMRP = 000007	TD\$STAR = 176372
BYTE31 = 000037	BYTE83 = 000123	MARLOD = 010000	Q\$LBD = 001000	TD\$TAR = 176362
BYTE32 = 000040	BYTE84 = 000124	MAROUT = 000002	Q\$LBDP = 001001	TD\$TDR = 176374
BYTE33 = 000041	BYTE85 = 000125	MAR.LO = 002000	Q\$LBP = 000001	TD\$TDW = 176364
BYTE34 = 000042	BYTE86 = 000126	MAR.OU = 000040	Q\$LDCD = 000003	TD\$AD = 000020
BYTE35 = 000043	BYTE87 = 000127	MBKALL = 001000	Q\$LDMD = 000004	T\$B = 000002
BYTE36 = 000044	BYTE88 = 000130	MBKCLK = 000400	Q\$LDPP = 002000	T\$B0 = 000010
BYTE37 = 000045	BYTE89 = 000131	MMARRD = 000100	Q\$LHP = 010000	T\$BSS = 100000
BYTE38 = 000046	BYTE9 = 000011	MMLEFT = 000002	Q\$MNC = 140000	T\$BT = 000020
BYTE39 = 000047	BYTE90 = 000132	MMOE = 000004	Q\$MR = 000052	T\$BTAR = 000030
BYTE4 = 000004	BYTE91 = 000133	MMJRT = 000010	Q\$MRP = 000040	T\$BTD = 002000
BYTE40 = 000050	BYTE92 = 000134	MNOBRE = 100000	Q\$MRP2 = 000240	T\$CD = 000100
BYTE41 = 000051	BYTE93 = 000135	MREN1 = 000001	Q\$MSC = 040000	T\$CLK = 002000

PPS.....MACRO-M1110 27-MAR-88 15:26 PAGE 10-2
SYMBOL TABLE

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

T\$DISK=000200	T\$SO=000001	WORD29=000072	WORD53=000152	WORD78=000234
T\$DRD=000004	T\$UBUS=100000	WORD3=000006	WORD54=000154	WORD79=000236
T\$MEM=010000	T\$1CLK=000400	WORD30=000074	WORD55=000156	WORD8=000020
T\$FSA=000000	T\$BEN=000020	WORD31=000076	WORD56=000160	WORD80=000240
T\$FSA=000004	UBD.IN=000020	WORD32=000100	WORD57=000162	WORD81=000242
T\$FSAC=000014	WORD0=000000	WORD33=000102	WORD58=000164	WORD82=000244
T\$FSB2=000010	WORD1=000002	WORD34=000104	WORD59=000166	WORD83=000246
T\$IB=000026	WORD10=000024	WORD35=000106	WORD6=000014	WORD84=000250
T\$IBAR=000024	WORD11=000026	WORD36=000110	WORD60=000170	WORD85=000252
T\$IBE=020000	WORD12=000030	WORD37=000112	WORD61=000172	WORD86=000254
T\$IBF=040000	WORD13=000032	WORD38=000114	WORD62=000174	WORD87=000256
T\$ICD=000040	WORD14=000034	WORD39=000116	WORD63=000176	WORD88=000260
T\$MODE=004000	WORD15=000036	WORD4=000010	WORD64=000200	WORD89=000262
T\$OB=000036	WORD16=000040	WORD40=000120	WORD65=000202	WORD9=000022
T\$OBE=004000	WORD17=000042	WORD41=000122	WORD66=000204	WORD90=000264
T\$OBF=010000	WORD18=000044	WORD42=000124	WORD67=000206	WORD91=000266
T\$OBRA=000034	WORD19=000046	WORD43=000126	WORD68=000210	WORD92=000270
T\$OBWA=000032	WORD2=000004	WORD44=000130	WORD69=000212	WORD93=000272
T\$OUTA=100000	WORD20=000050	WORD45=000132	WORD7=000016	WORD94=000274
T\$RBD=000200	WORD21=000052	WORD46=000134	WORD70=000214	WORD95=000276
T\$RNB=000040	WORD22=000054	WORD47=000136	WORD71=000216	WORD96=000300
T\$RSET=040000	WORD23=000056	WORD48=000140	WORD72=000220	WORD97=000302
T\$SC=000022	WORD24=000060	WORD49=000142	WORD73=000222	WORD98=000304
T\$SCLK=020000	WORD25=000062	WORD5=000012	WORD74=000224	WORD99=000306
T\$SEG1=000000	WORD26=000064	WORD50=000144	WORD75=000226	WRDVAL=000310
T\$SEG2=000001	WORD27=000066	WORD51=000146	WORD76=000230	XTREAD=001000
T\$SEG3=000002	WORD28=000070	WORD52=000150	WORD77=000232	XTURTE=000400

. ABS. 000000 000
000000 001
PPS 000464 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3251 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:43
PPS,PPS/SP=[20,1]IM.[20,1]PPS

Approved For Release 2005/07/10 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

58 000044 000000      CDDAT: .WORD 0      ;DATA FIELD FOR TRANSFER TO CD
59 000046 000000      CODE: .WORD 0      ;CODE FOR MEMORY SELECTION
60
61
62
63
64
65
66
67
68
69 000050      LD4::
70 000050
71 000054 103004      CALL . FIND      ;LOCATE MEMORY MNEMONIC IN COMMAND LINE
72 000056      BCC . 1$      ;OK, CONTINUE
73 000062 000167 001672      CALL . ERR4      ;MISSING OPERAND
74
75      JMP . LD4X      ;EXIT
76
77
78
79 000066 012700 000010      ;MATCH THE MNEMONIC IN THE COMMAND LINE AGAINST THE TABLE
80 000072 012702 000000      OF VALID MNEMONICS.
81 000076      1$:
82 000102 103004      MOV . #LD4LN,R0      ;NUMBER OF TABLE ENTRIES
83 000110 000167 001644      MOV . #LD4TBL,R2      ;POINT TO TABLE
84
85      CALL . SCAN      ;MATCH AGAINST COMMAND LINE
86
87      BCC . 2$      ;MATCH WAS MADE
88
89      CALL . ERR7      ;INVALID MEMORY MNEMONIC
90      JMP . LD4X
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106 000114 010167 000000G.      ;SAVE POINTER TO ROUTINE ASSOCIATED WITH THE MEMORY MNEMONIC
107 000120      ;SCAN THE REMAINDER OF THE COMMAND LINE FOR CONTROL INFORMATION
108 000124 103004
109 000126
110 000132 000167 001622
111 000136
112 000142 103004
113 000144
114 000150 000167 001604

```

1. START ADDRESS ONLY. REPEAT PROMPT FOR EACH MEMORY LOCATION FROM THE START ADDRESS FORWARD. NB. END OF MEMORY (AS DETECTED BY THE QMAIN SUBROUTINE PDATA) OR A <CR> RESPONSE TO THE PROMPT WILL TERMINATE THE LOAD.
PP>LD QW 0

2. START ADDRESS, LOOP INDICATOR, LOOP ON THE LOADING OF THIS ONE MEMORY LOCATION ONLY.
PP>LD QW 0 L

3. START ADDRESS, END ADDRESS, NO LOOP, ISSUE ONLY ONE PROMPT AND FILL MEMORY (BETWEEN START AND END ADDRESSES) WITH THIS VALUE.
PP>LD QW 0 7

4. START ADDRESS, END ADDRESS, LOOP INDICATOR, ISSUE ONLY ONE PROMPT AND LOOP ON THE LOADING OF MEMORY (BETWEEN START AND END ADDRESSES) WITH THIS VALUE.
PP>LD QW 0 7 L

```

2$:      MOV . R1,RTNPT      ;SAVE POINTER
      CALL . FIND      ;LOCATE START ADDRESS IN COMMAND LINE
      BCC . 3$      ;OK, CONTINUE
      CALL . ERR4      ;MISSING OPERAND
      JMP . LD4X      ;EXIT
3$:      CALL . PACK      ;CONVERT COMMAND LINE VALUE TO BINARY
      BCC . 4$      ;CONVERSION SUCCESSFUL
      CALL . ERR5      ;INVALID NUMERIC VALUE
      JMP . LD4X

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

115
116 000154 016767 000000G-000000G-4$: MOV. BINWD,MSTR1. :SAVE-LOADING-START-ADDRESS.
117 000162 016767 000000G-000000G- MOV. BINWD,MSTR2. :SAVE-IT-TWICE-(FOR-REFRESH-ON-LOOP-7$).
118 000170 012767 177777 000000G- MOV. #-1,MEND. :UNT-END-ADDRESS.
119
120
121 : START-ADDRESS-HAS-BEEN-FOUND. SCAN-FOR-END-ADDRESS-OR-LOOP
122 : INDICATOR-(CONDITIONS-2, 3 ABOVE). IF-THERE-IS-NOTHING.
123 : FURTHER-IN-THE-COMMAND-LINE, CONDITION-1-IS-IN-EFFECT.
124 000176
125 000202 103004 CALL. FIND :SCAN-COMMAND-LINE.
126 000204 052767 000000G-000000G- BCC. 5$ :SOMETHING-THERE.
127 000212 000437 BIS. *RP,BASE. :SIGNAL-TO-REPEAT-PROMPT.
128 BR. 9$ :JUMP-TO-RTN.
129
130 000214 122711 000114 5$: CMPB. #'L,(R1) :LOOP-INDICATOR.
131 000220 001006 BNE. 6$ :NO-MUST-BE-UPPER-ADDRESS.
132 000222 016767 000000G-000000G- MOV. MSTR1,MEND. :SET-END-ADDR.=START-ADDR.
133 000234 000426 CALL. HANG :HOW-TO-STOP-LOOP.
134 BR. 9$ :JUMP-TO-RTN.
135
136 000236 6$: CALL. PACK :CONVERT-UPPER-ADDRESS.
137 000242 103004 BCC. 7$ :OK-CONTINUE.
138 000244 000244 CALL. ERR5 :INVALID-NUMERIC.
139 000250 000167 001504 JMP. LD4X :EXIT.
140
141 :
142 : SAVE-END-ADDRESS-(BINARY)
143 : CHECK-FOR-LOOP-INDICATOR-AFTER-END-ADDRESS (CONDITION-4)
144 000254 016767 000000G-000000G-7$: MOV. BINWD,MEND. :SET-UP-END-ADDRESS.
145 000262 CALL. FIND :CHECK-FOR-LOOP-INDICATOR.
146 000266 103411 BCS. 9$ :NO-LOOP.
147 000270 122711 000114 CMPB. #'L,(R1) :CORRECT-INDICATOR.
148 000274 001404 BEQ. 8$ :YES-CONTINUE.
149 000276 CALL. ERR11 :LOOP-OPTION-ERROR
150 000302 000167 001452 JMP. LD4X
151 000306 8$: CALL. HANG :HOW-TO-STOP-LOOP.
152
153 000312 016701 000000G- 9$: MOV. RTNPT,R1 :POINT-TO-ROUTINE.
154 000316 000171 000000 JMP. @ (R1) :JUMP-TO-ROUTINE.
155
156 :
157 : LOAD-QEX-WINDOW-MEMORY
158 : LOAD-QEX-LOCATION-MEMORY.
159 :
160 000322 012767 000042 177516 LD4QW:: MOV. #Q$QW, CODE. :SET-MEMORY-SELECT-CODE.=WINDOW.
161 000330 000403 BR. QEX.
162 000332 012767 000043 177506 LD4QL:: MOV. #Q$QL, CODE. :SET-MEMORY-SELECT-CODE.=LOCATION.
163
164 000340 016746 000000G- QEX: MOV. QXHIGH,-(SP) :SUPPLY-UPPER-MEMORY-LIMIT.
165 000344 016746 000000G- MOV. QXLOW,-(SP) :LOWER-LIMIT.
166 000350 CALL. BUSET. :PREPARE-FOR-LOAD.
167 000354 103002 BCC. 1$ :OK-CONTINUE.
168 000356 000167 001376 JMP. LD4X :ERROR-EXIT.
169
170 :
171 : WRITE-DATA-FOR-QEX-INTO-CP-DATA-MEMORY. SKIP-CD-WORDS-0, 1.
: (RESERVED-FOR-WRITE-FLAG-AND-TRANSFER-COUNT-SEE-BELOW).

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

172.      ;      WRITE QEX DATA IN THE FORM ADDR, DATA, ADDR, DATA ...
173.      ;      WHEN THE LOADING OF CD IS COMPLETED, CALL THE SUBROUTINE
174.      ;      QXLOAD WHICH IN TURN STARTS UP THE MICROCODE TO LOAD THE
175.      ;      QEX.
176.
177 000362 012767 000002 177450 1$: MOV. #2, CDADD.      ;WRITE INTO CD STARTING AT LOC 2.
178 000370 005067 177446      CLR. QXCNT.      ;CLEAR QEX ADDR/DATA PAIRS COUNT.
179 000374 005046      CLR. -(SP)      ;START MICROCODE AT 0
180 000376      CALL. SEQCS.
181 000402 005046      CLR. -(SP)      ;RESET BR INHIBIT.
182 000404      CALL. CPCR.
183 000410 012746 000377      MOV. #377, -(SP)      ;SET MRP MICRO ADDRESS - X'FF' (JUMP SELF)
184 000414      CALL. SEQMM.
185 000420 005046      CLR. -(SP)      ;RESET BR INHIBIT.
186 000422      CALL. MRPCR.
187 000426 012767 001000 176422      MOV. #Q$REBK, QR$CR2      ;RE-ARM INTERRUPTS.
188 000434 012767 120000 176422      MOV. #<Q$SM+Q$ENOP>, QR$CR2.      ;SET SEARCH MODE + ENABLE INTERRUPTS.
189 000442 012746 000360      MOV. #Q$CSEL, -(SP)      ;CLEAR ALL SELECTIONS.
190 000446 052716 001001      BIS. #<Q$LBD+Q$LBP>, (SP)      ;CLEAR DRIVE AND PULSE.
191 000452 052716 030000      BIS. #Q$CNC, (SP)      ;CLEAR CP NO-CLOCK.
192 000456 005046      CLR. -(SP)      ;SET NOTHING.
193 000460      CALL. CSR1.
194
195 000464 032767 000000G 000000G.      BIT. #RP, BASE.      ;REPEAT PROMPT ?
196 000472 001430      BEQ. 4$      ;NO.
197 000474      CALL. PDATA.      ;READ DATA FROM COMMAND LINE.
198 000500 103421      BCS. 3$      ;END OF MEMORY - LOAD QEX.
199 000502 102420      BVS. 3$      ;<CR> RESPONSE - LOAD QEX.
200 000504 016767 000000G 177332      MOV. MSTR2, CDDAT      ;WRITE QEX ADDR TO CP DATA MEMORY.
201 000512      CALL. LOADCD.      ;LOAD ONE WORD.
202 000516 016767 000000G 177320      MOV. DATA1, CDDAT.      ;WRITE QEX DATA TO CP DATA MEMORY.
203 000524      CALL. LOADCD.
204 000530 005267 177306      INC. QXCNT.      ;COUNT NUMBER OF QEX ADDR/DATA PAIRS.
205 000534 066767 000000G 000000G.      ADD. INCVAL, MSTR2.      ;BUMP LOAD ADDRESS.
206 000542 000754      BR. 2$      ;REPEAT.
207
208 000544      ;
209 000550 000167 001204      CALL. LOADQX.      ;LOAD QEX.
210      JMP. LD4X      ;EXIT.
211
212      ;      PROMPT ONCE, THEN FILL MEMORY.
213
213 000554      ;
214 000560 103401      CALL. PDATA.      ;PROMPT.
215 000562 102004      BCS. 5$      ;END OF MEMORY OR ERROR.
216 000564      BVC. 6$      ;NORMAL CONTINUATION.
217 000570 000167 001164      CALL. LOADQX.      ;LOAD QEX.
218 000574 016767 000000G 177242      JMP. LD4X
219 000582      MOV. MSTR2, CDDAT.      ;WRITE QEX ADDR TO CP DATA MEMORY.
220 000606 016767 000000G 177230      CALL. LOADCD.      ;LOAD ONE WORD.
221 000614      MOV. DATA1, CDDAT.      ;WRITE QEX DATA TO CP DATA MEMORY.
222 000620 005267 177216      CALL. LOADCD.
223 000624 066767 000000G 000000G.      INC. QXCNT.      ;INC QEX ADDR/DATA PAIRS COUNT.
224 000632 026767 000000G 000000G.      ADD. INCVAL, MSTR2.      ;ADVANCE ADDRESS.
225 000640 101755      CMP. MSTR2, MEND.      ;HAS UPPER MEMORY LIMIT BEEN REACHED.
226 000642      BLOS. 6$      ;NO, CONTINUE.
227 000646 032767 000000G 000000G.      CALL. LOADQX.      ;LOAD QEX.
228 000654 001372      BIT. #LOOP, BASE.      ;LOOP ON ?
      BNE. 7$      ;YES, CONTINUE.

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

229 000656 000167 001076 JMP LD4X :ELSE EXIT
230 :
231 :
232 : FAL POINTER MEMORY
233 : FAL COUNTER MEMORY
234 :
235 :
236 000662 012767 000046 177156 LD4FP: MOV #Q$FP, CODE :SET MEMORY SELECT CODE = POINTER
237 000670 000403 BR FAL
238 000672 012767 000045 177146 LD4FC: MOV #Q$FC, CODE :SET MEMORY SELECT CODE = COUNTER
239 :
240 000700 016746 000000G FAL: MOV FAHIGH, -(SP) :SUPPLY UPPER MEMORY LIMIT
241 000704 016746 000000G MOV FALOW, -(SP) :LOWER LIMIT
242 000710 CALL BUFSET :PREPARE FOR LOAD
243 000714 103002 BCC 1$ :OK, CONTINUE
244 000716 000167 001036 JMP LD4X :ERROR, EXIT
245 :
246 000722 032767 000000G 000000G 1$: BIT #RP, BASE :REPEAT PROMPT
247 000730 001436 BEQ 5$ :NO, ONCE ONLY
248 000732 2$: CALL PDATA :READ DATA FROM COMMAND LINE
249 000736 103401 BCS 3$ :END OF MEMORY
250 000740 102006 BVC 4$ :NO <CR> RESPONSE, CONTINUE
251 000742 012746 077777 3$: MOV #077777, -(SP) :VALUE FOR QCL POINTER
252 000746 CALL STOP :LOAD QCL POINTER
253 000752 000167 001002 JMP LD4X
254 :
255 000756 016746 000000G 4$: MOV MSTR2, -(SP) :LOAD ADDR INTO QCL POINTER
256 000762 CALL STOP
257 000766 016746 177054 MOV CODE, -(SP) :SELECT MEMORY
258 000772 CALL PPCR :WRITE SELECTION TO CONTROL REG
259 000776 016746 000000G MOV DATA1, -(SP) :SEND DATA WORD TO FAL MEMORY
260 001002 CALL LBPP
261 001006 012746 000040 MOV #Q$CLR, -(SP)
262 001012 CALL PPCR
263 :
264 001016 066767 000000G 000000G ADD INCVAL, MSTR2 :BUMP ADDRESS
265 001024 000742 BR 2$ :REPEAT
266 :
267 :
268 : PROMPT ONCE THEN FILL MEMORY
269 001026 5$: CALL PDATA :PROMPT
270 001032 103401 BCS 6$ :END OF MEMORY
271 001034 102006 BVC 7$ :NO <CR> RESPONSE, CONTINUE
272 001036 012746 077777 6$: MOV #077777, -(SP) :VALUE FOR QCL POINTER
273 001042 CALL STOP :LOAD QCL POINTER
274 001046 000167 000706 JMP LD4X
275 :
276 001052 016746 000000G 7$: MOV MSTR2, -(SP) :LOAD ADDR INTO QCL POINTER
277 001056 CALL STOP
278 001062 016746 176760 MOV CODE, -(SP) :SELECT MEMORY
279 001066 CALL PPCR :WRITE SELECTION TO CONTROL REG
280 001072 016746 000000G MOV DATA1, -(SP) :SEND DATA WORD TO FAL MEMORY
281 001076 CALL LBPP
282 001102 012746 000040 MOV #Q$CLR, -(SP) :CLEAR PPS
283 001106 CALL PPCR
284 :
285 001112 066767 000000G 000000G ADD INCVAL, MSTR2

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

286 001120 026767 000000G-000000G- CMP- MSTR2,MEND- ;HAS-UPPER-MEMORY-LIMIT-BEEN-REACHED-
287 001126 101751 BLOS- 7$ ;NO, CONTINUE-
288 001130 332767 000000G-000000G- BIT- #LOOP,BASE- ;LOOP-ON-
289 001136 001006 BNE- 8$ ;YES, CONTINUE-
290 001140 012746 077777 MOV- #077777,-(SP) ;VALUE-FOR-QCL-POINTER-
291 001144 CALL- STOP ;LOAD-QCL-POINTER-
292 001150 000167 000604 JMP- LD4X
293 001154 016767 000000G-000000G-0$ MOV- MSTR,MSTR2- ;REINITIALIZE-ADDRESS-
294 001162 000133 BR- 7$
295 ;
296 ;
297 ; LOAD-QLB-REFERENCE-PAGE-
298 ;
299 ;
300 001164 LD4QR::
301 001164 016746 000000G- MOV- LHHIGH,-(SP) ;SUPPLY-UPPER-MEMORY-LIMIT-
302 001170 016746 000000G- MOV- LHLOW,-(SP) ;LOWER-LIMIT-
303 001174 CALL- BUFSET- ;PREPARE-FOR-LOAD-
304 001200 103002 BCC- 1$ ;OK, CONTINUE-
305 001202 000167 000552 JMP- LD4X ;EPROP-EXIT-
306 ;
307 001206 032767 000000G-000000G-1$ BIT- #RP,BASE- ;REPEAT-PROMPT-
308 001214 001436 BEQ- 5$ ;NO, ONCE-ONLY-
309 001216 2$ CALL- PDATA- ;READ-DATA-FROM-COMMAND-LINE-
310 001222 103401 BCS- 3$ ;END-OF-MEMORY-
311 001224 102006 BVC- 4$ ;NO-⟨CR⟩-RESPONSE, CONTINUE-
312 001226 012746 000013 3$ MOV- #Q$QHLT,-(SP) ;HALT-CODE
313 001232 CALL- PPCR
314 001236 000167 000516 JMP- LD4X
315 ;
316 001242 012746 000053 4$ MOV- #Q$QLA,-(SP) ;ADDRESS-SELECT-FOR-QLB-PAGE-
317 001246 CALL- PPCR ;SEND-TO-PP-CONTROL-REG-
318 001252 016746 000000G- MOV- MSTR2,-(SP) ;ACTUAL-ADDRESS-
319 001256 CALL- LBPP ;SEND-TO-PP-
320 001262 012746 000001 MOV- #Q$QLR,-(SP) ;SELECT-QLB-REF-MEMORY-
321 001266 CALL- PPCR
322 001272 016746 000000G- MOV- DATA1,-(SP) ;DATA-WORD-FOR-MEMORY-
323 001276 CALL- LBPP ;SEND-DATA-TO-PPS-
324 ;
325 001302 066767 000000G-000000G- ADD- INCVAL,MSTR2- ;BUMP-ADDRESS-
326 001310 000742 BR- 2$ ;REPEAT-
327 ;
328 ;
329 ; PROMPT-ONCE-THEN-FILL-MEMORY-
330 001312 5$ CALL- PDATA- ;PROMPT-
331 001316 103401 BCS- 6$ ;END-OF-MEMORY-
332 001320 102006 BVC- 7$ ;NO-⟨CR⟩-RESPONSE, CONTINUE-
333 001322 012746 000013 6$ MOV- #Q$QHLT,-(SP) ;HALT-CODE
334 001326 CALL- PPCR
335 001332 000167 000422 JMP- LD4X
336 ;
337 001336 012746 000053 7$ MOV- #Q$QLA,-(SP) ;ADDRESS-SELECT-FOR-QLB-PAGE-
338 001342 CALL- PPCR ;SEND-TO-PP-CONTROL-REG-
339 001346 016746 000000G- MOV- MSTR2,-(SP) ;ACTUAL-ADDRESS-
340 001352 CALL- LBPP ;SEND-TO-PP-
341 001356 012746 000001 MOV- #Q$QLR,-(SP) ;SELECT-QLB-REF-MEMORY-
342 001362 CALL- PPCR

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

343 001366 016746 000000G..... MOV. DATA1,-(SP) ;DATA WORD FOR MEMORY.
344 001372. CALL. LBPP ;SEND DATA TO PPS.
345
346 001376 066767 000000G-000000G. ADD. INCVAL,MSTR2. ;ADVANCE ADDRESS.
347 001404 026767 000000G-000000G. CMP. MSTR2,MEND. ;HAS UPPER MEMORY LIMIT BEEN REACHED.
348 001412. 101751 BLOS. 7$ ;NO. CONTINUE.
349 001414 032767 000000G-000000G. BIT. #LOOP,BASE. ;LOOP ON.
350 001422. 001006 BNE. 8$ ;YES. CONTINUE.
351 001434 012746 000013 MOV. #Q$QHLT,-(SP) ;HALT CODE
352 001430 CALL. PPCR ;WRITE TO PPS CONTROL REGISTER
353 001434 000167 000320 JMP. LD4X
354 001440 016767 000000G-000000G-8$: MOV. MSTR2,MSTR2. ;REINITIALIZE ADDRESS.
355 001446 000733 BR. 7$
356
357
358
359
360
361 001450 LD4Q0::
362 001450 012746 000000 MOV. #0,-(SP) ;SELECT PAGE 0
363 001454 000405 BR. PGSEL.
364 001456 LD4Q1::
365 001456 012746 000001 MOV. #1,-(SP) ;SELECT PAGE 1
366 001452. 000402. BR. PGSEL.
367 001464 LD4Q2::
368 001464 012746 000002 MOV. #2,-(SP) ;SELECT PAGE 2.
369
370 PGSEL:: CALL. SELPG. ;SELECT A QLB PAGE.
371 001474 016746 000000G. MOV. LHHIGH,-(SP) ;SUPPLY UPPER MEMORY LIMIT.
372 001500 016746 000000G. MOV. LHLOW,-(SP) ;LOWER LIMIT.
373 001504 CALL. BUFSET. ;PREPARE FOR LOAD.
374 001510 103002. BCC. 1$ ;OK. CONTINUE.
375 001512. 000167 000242 JMP. LD4X ;ERROR, EXIT.
376
377 001516 032767 000000G-000000G-1$: BIT. #RP,BASE. ;REPEAT PROMPT.
378 001524 001436 BEQ. 5$ ;NO. ONCE ONLY.
379 001526 2$: CALL. PDATA. ;READ DATA FROM COMMAND LINE.
380 001532. 103401 BCS. 3$ ;END OF MEMORY.
381 001534 102006 BVC. 4$ ;NO <CR> RESPONSE. CONTINUE.
382 001536 012746 000013 MOV. #Q$QHLT,-(SP) ;HALT CODE
383 001542. CALL. PPCR ;WRITE TO PPS CONTROL REGISTER
384 001546 000167 000206 JMP. LD4X
385
386 001552. 012746 000053 4$: MOV. #Q$QLA,-(SP) ;ADDRESS SELECT FOR QLB PAGE
387 001556 CALL. PPCR ;SEND TO PP CONTROL REG.
388 001562. 016746 000000G. MOV. MSTR2,-(SP) ;ACTUAL ADDRESS.
389 001566 CALL. LBPP ;SEND TO PP.
390 001572. 012746 000054 MOV. #Q$QLB,-(SP) ;SELECT QLB REF MEMORY.
391 001576 CALL. PPCR
392 001602. 016746 000000G. MOV. DATA1,-(SP) ;DATA WORD FOR MEMORY.
393 001606 CALL. LBPP ;SEND DATA TO PPS.
394
395 001612. 066767 000000G-000000G. ADD. INCVAL,MSTR2. ;BUMP ADDRESS.
396 001620 000742. BR. 2$
397
398
399

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

400 001622.		5\$:	CALL	PDATA.	:PROMPT.
401 001626	103401		BCS.	6\$:END OF MEMORY.
402 001630	102006		BVC.	7\$:NO. CR> RESPONSE. CONTINUE.
403 001632	012746	000013	6\$:	MOV.	#Q\$OHLT, -(SP)
404 001636			CALL	PPCR	:HALT CODE
405 001642	000167	000112	JMP.	LD4X	:WRITE TO PPS CONTROL REGISTER
406			:		
407 001646	012746	000053	7\$:	MOV.	#Q\$OLA, -(SP)
408 001652			CALL	PPCR	:ADDRESS SELECT FOR QLB PAGE
409 001656	016746	000000G.	MOV.	MSTR2, -(SP)	:SEND TO PP CONTROL REG
410 001662			CALL	LBPP	:ACTUAL ADDRESS
411 001666	012746	000054	MOV.	#Q\$QLB, -(SP)	:SEND TO PP
412 001672			CALL	PPCR	:SELECT QLB MEMORY
413 001676	016746	000000G.	MOV.	DATA1, -(SP)	:DATA WORD FOR MEMORY
414 001702			CALL	LBPP	:SEND DATA TO PPS
415			:		
416 001706	066767	000000G. 000000G.	ADD.	INCVL, MSTR2.	:ADVANCE ADDRESS
417 001714	026767	000000G. 000000G.	CMP.	MSTR2, MEND.	:HAS UPPER MEMORY LIMIT BEEN REACHED
418 001722	101751		BLOS.	7\$:NO, CONTINUE
419 001724	032767	000000G. 000000G.	BIT.	#LOOP, BASE.	:LOOP ON
420 001732	001006		BNE.	8\$:YES, CONTINUE
421 001734	012746	000013	MOV.	#Q\$OHLT, -(SP)	:HALT CODE
422 001740			CALL	PPCR	:WRITE TO PPS CONTROL REGISTER
423 001744	000167	000010	JMP.	LD4X	
424 001750	016767	000000G. 000000G. 8\$:	MOV.	MSTR1, MSTR2.	:REINITIALIZE ADDRESS
425 001756	000733		BR.	7\$	
426			:		
427			:		
428 001760			LD4X:		
429 001760	012746	000040	MOV.	#Q\$CLR, -(SP)	:CLEAR PPS
430 001764			CALL	PPCR	:WRITE TO CONTROL REG
431 001770	042767	000000G. 000000G.	BIC.	#RP, BASE.	:CLEAR PROMPT FLAG
432 001776			CALL	KILL	:KILL AST. (IF THERE WAS ONE)
433 002002	000167	000000G.	JMP.	PPSXX.	

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

435      ;
436      ;
437      ;      LOAD QEX MEMORY.
438      ;
439      ;
440      ;      FIRST WRITE THE FLAG ('1' = WRITE) AND THE TRANSFER COUNT
441      ;      INTO THE FIRST TWO LOCATIONS IN CP DATA MEMORY.
442      ;
443      ;      LOADQX:
444      002006      005767      176030      TST      QXCNT      ;ANYTHING TO LOAD
445      002012      001507      BEQ      1$      ;NO, SHUT DOWN AND EXIT
446      ;
447      002014      005067      176020      CLR      CDADD      ;WRITE FLAG TO CD ADDR ZERO
448      002020      012767      000001      176016      MOV      #1, CDDAT      ;FLAG FOR 'WRITE QEX'
449      002026      CALL      LOADCD
450      002032      016767      176004      176004      MOV      QXCNT, CDDAT      ;WRITE QEX ADDR/DATA PAIRS COUNT TO CD ADDR 1
451      002040      CALL      LOADCD
452      ;
453      002044      005046      CLR      -(SP)      ;CLEAR NOTHING IN CSR1
454      002046      012746      176000      MOV      #0$NCLK, -(SP)      ;SET NO-CLOCKS
455      002052      CALL      CSR1
456      002056      005067      176422      CLR      QR$CR2      ;SET LOAD MODE
457      ;
458      ;      SELECT MEMORY TO BE LOADED (WINDOW OR LOCATION).
459      ;      START THE MICROCODE THAT DOES THE LOADING OF THE QEX.
460      ;
461      002062      016746      175760      MOV      CODE, -(SP)      ;SELECT WINDOW OR LOCATION MEMORY
462      002066      CALL      PPCR
463      002072      012746      001760      MOV      #1760, -(SP)      ;START CP MICROCODE AT X'3F0'
464      002076      CALL      SEQCS
465      002102      005046      CLR      -(SP)      ;RESET BR INHIBIT
466      002104      CALL      CPCR
467      002110      012746      000377      MOV      #377, -(SP)      ;SET MRP MICRO ADDRESS = X'FF' (JUMP SELF)
468      002114      CALL      SEQMM
469      002120      005046      CLR      -(SP)      ;RESET BR INHIBIT
470      002122      CALL      MRPCR
471      002126      012767      001000      176422      MOV      #Q$REBK, QR$CR2      ;RE-ARM INTERRUPTS
472      002134      012767      022000      176422      MOV      #<Q$LDPP+Q$ENOP>, QR$CR2      ;SET LOAD PPS MODE + ENABLE INTERRUPTS
473      002142      012746      000360      MOV      #Q$CSEL, -(SP)      ;CLEAR ALL SELECTIONS
474      002146      052716      001001      BIS      #<Q$LBD+Q$LBP>, (SP)      ;CLEAR DRIVE AND PULSE
475      002152      052716      176000      BIS      #Q$NCLK, (SP)      ;CLEAR ALL NO-CLOCKS
476      002156      005046      CLR      -(SP)      ;SET NOTHING
477      002160      CALL      CSR1
478      ;
479      ;      WAIT FOR INTERRUPT FROM CP.
480      ;      USE DMA INTERRUPT (SEE CP MICROCODE FOR QEX)
481      ;
482      002164      WTSE$S      #EFN.3
483      ;
484      002176      CLEF$S      #EFN.3
485      ;
486      ;      RE-ARM INTERRUPTS.
487      ;
488      002210      012767      000400      176422      MOV      #<Q$SM+Q$CHB>, QR$CR2      ;CLEAR INTERRUPT (USE INT BUFFER INT)
489      002216      012767      001000      176422      MOV      #<Q$SM+Q$REBK>, QR$CR2      ;RE-ARM
490      002224      012767      000000      176422      MOV      #<Q$SM+Q$ENBK+Q$ENOP>, QR$CR2      ;ENABLE
491      ;

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

492	002232	005046	1\$:	CLR	-(SP)	:CLEAR NOTHING IN CSR1
493	002234	012746		MOV	#Q\$NCLK, -(SP)	:SET NO-CLOCKS
494	002240			CALL	CSR1	
495	002244	005067		CLR	QR\$CR2	:SET LOAD MODE
496	002250			RETURN		

Approved For Release 2005/07/11 : CIA-RDP85-00514R000200020001-3

ALUCKE = 040000
ALUOE = 004000
A01 = 010000
BASE = ***** GX
BINWD = ***** GX
BITVAL = 000000
BIT0 = 000001
BIT1 = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2 = 000004
BIT3 = 000010
BIT4 = 000020
BIT5 = 000040
BIT6 = 000100
BIT7 = 000200
BIT8 = 000400
BIT9 = 001000
BUFSET = ***** GX
BYTE0 = 000000
BYTE1 = 000001
BYTE10 = 000012
BYTE11 = 000013
BYTE12 = 000014
BYTE13 = 000015
BYTE14 = 000016
BYTE15 = 000017
BYTE16 = 000020
BYTE17 = 000021
BYTE18 = 000022
BYTE19 = 000023
BYTE2 = 000002
BYTE20 = 000024
BYTE21 = 000025
BYTE22 = 000026
BYTE23 = 000027
BYTE24 = 000030
BYTE25 = 000031
BYTE26 = 000032
BYTE27 = 000033
BYTE28 = 000034
BYTE29 = 000035
BYTE3 = 000003
BYTE30 = 000036
BYTE31 = 000037
BYTE32 = 000040
BYTE33 = 000041
BYTE34 = 000042
BYTE35 = 000043
BYTE36 = 000044
BYTE37 = 000045
BYTE38 = 000046
BYTE39 = 000047
BYTE4 = 000004
BYTE40 = 000050
BYTE41 = 000051
BYTE42 = 000052
BYTE43 = 000053
BYTE44 = 000054
BYTE45 = 000055
BYTE46 = 000056
BYTE47 = 000057
BYTE48 = 000060
BYTE49 = 000061
BYTE5 = 000005
BYTE50 = 000062
BYTE51 = 000063
BYTE52 = 000064
BYTE53 = 000065
BYTE54 = 000066
BYTE55 = 000067
BYTE56 = 000070
BYTE57 = 000071
BYTE58 = 000072
BYTE59 = 000073
BYTE6 = 000006
BYTE60 = 000074
BYTE61 = 000075
BYTE62 = 000076
BYTE63 = 000077
BYTE64 = 000100
BYTE65 = 000101
BYTE66 = 000102
BYTE67 = 000103
BYTE68 = 000104
BYTE69 = 000105
BYTE7 = 000007
BYTE70 = 000106
BYTE71 = 000107
BYTE72 = 000110
BYTE73 = 000111
BYTE74 = 000112
BYTE75 = 000113
BYTE76 = 000114
BYTE77 = 000115
BYTE78 = 000116
BYTE79 = 000117
BYTE8 = 000008
BYTE80 = 000120
BYTE81 = 000121
BYTE82 = 000122
BYTE83 = 000123
BYTE84 = 000124
BYTE85 = 000125
BYTE86 = 000126
BYTE87 = 000127
BYTE88 = 000130
BYTE89 = 000131
BYTE9 = 000009
BYTE90 = 000132
BYTE91 = 000133
BYTE92 = 000134
BYTE93 = 000135
BYTE94 = 000136
BYTE95 = 000137
BYTE96 = 000140
BYTE97 = 000141
BYTE98 = 000142
BYTE99 = 000143
BYTVAL = 000144
CBKALL = 001000
CBKCLK = 000400
CDADD = 000040R
CDDAT = 000044R
CNOBRE = 100000
CODE = 000046R
CPCCEN = 010000
CPCR = ***** GX
CPREAD = 040000
CPWRTE = 020000
CSADPD = 000004
CSEQCI = 100000
CSOE = 000040
CSR1 = ***** GX
CSWRTE = 000100
DATA1 = ***** GX
DBR RD = 000001
DBRCPD = 001457
DBSPT = 000026
DBSTPC = 000023
DISPGS = 100000
DMAAUR = 000005
DMARRD = 000003
DMARWR = 000004
EFN3 = ***** GX
ENBR = 010000
ERR11 = ***** GX
ERR4 = ***** GX
ERR5 = ***** GX
ERR7 = ***** GX
FAHIGH = ***** GX
FAL = 000700R
FALOW = ***** GX
FIND = ***** GX
HANG = ***** GX
INCVAL = ***** GX
KILL = ***** GX
LBPP = ***** GX
LD4 = 000050RG
LD4FC = 000672RG
LD4FP = 000662RG
LD4LN = 000010 G
LD4QL = 000332RG
LD4QR = 001164RG
LD4QW = 000322RG
LD4QW = 001450RG
LD4Q1 = 001456RG
LD4Q2 = 001464RG
LD4TBL = 000000RG
LD4X = 001760R
LHHIGH = ***** GX
LHLOW = ***** GX
LOADCD = 002252R
LOADQX = 002006R
LOC EN = 000100
LOC WA = 040000
LOC WB = 100000
LOOP = ***** GX
MAREN1 = 000001
002 MAREN2 = 004000
002 MARLOD = 010000
MAROUT = 000002
002 MAR LO = 002000
MAR QU = 000040
MBKALL = 001000
MBKCLK = 000400
MHND = ***** GX
MMADR = 000100
MMLEFT = 000002
MMOE = 000004
MMWRTE = 000010
MNOBRE = 100000
MREN1 = 000001
MREN2 = 020000
MRPCR = ***** GX
MSTR = ***** GX
MSTR2 = ***** GX
MSYN = 000040
N = 000144
PACK = ***** GX
PDATA = ***** GX
PGSEL = 001470R
PLB = 000010
PLC = 000020
PLD = 000030
PLRWR = 000200
PLR EN = 000200
PPCR = ***** GX
002 PPSW = ***** GX
QEX = 000340R
QRSCR1 = 176420
QRSCR2 = 176422
QRLBR = 176424
QXCNT = 000042R
QXHIGH = ***** GX
002 QXLOW = ***** GX
002 QATTN = 000100
002 QBLD = 000001
QCCCP = 000040
002 QCHB = 000400
002 QCHRL = 000200
002 QCLR = 000040
002 QCNC = 030000
002 QCP = 000060
QCPCC = 000010
QCP2 = 000260
QCSCL = 010000
QCCSEL = 000360
QCCSET = 000002
QCCSP = 020000
QCDMA = 000001
QENBK = 040000
QENOP = 020000
QFAL = 004000
QFC = 000045
QFO = 000044
QFP = 000046
QHBF = 000002
QICP = 000006
QIHB = 000003
QIHRL = 000002
QIMRP = 000007
QLBD = 001000
QLBDP = 001001
QLBP = 000001
QLDCD = 000003
QLDMD = 000004
QLDPP = 002000
QLHP = 010000
QMNC = 140000
QMR = 000052
QMRP = 000040
QMRP2 = 000240
QMNC = 040000
QMSET = 000004
QMSP = 100000
QMCLK = 176000
QPP = 000100
002 QPPSW = 000320
QPLB = 000300
QOHLT = 000013
QQL = 000043
QQLA = 000053
QQLB = 000054
QQLB = 000001
QQLW = 000042
002 QRDCD = 000005
QRDMD = 000006
QREBK = 001000
QRNC = 006000
002 QRSC = 004000
QRSET = 000010
QSM = 100000
QSP = 000120
QSP2 = 000340
RGD EN = 000200
RGD VA = 020000
RP = ***** GX
RTNPT = ***** GX
SCAN = ***** GX
SELPG = ***** GX

PPLD---MACRO-M1110 27-MAR-80 15:23 PAGE 7-2
SYMBOL TABLE

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

SEQCS = .***** GX.	T\$DRD = .000004	WORD11 = .000026	WORD42 = .000124	WORD72 = .000220
SEQMM = .***** GX.	T\$MEM = .010000	WORD12 = .000030	WORD43 = .000126	WORD73 = .000222
SEQ.CI = .000010	T\$FSA = .000000	WORD13 = .000032	WORD44 = .000130	WORD74 = .000224
STQP = .***** GX.	T\$FSAB = .000004	WORD14 = .000034	WORD45 = .000132	WORD75 = .000226
S\$CLR = .000000	T\$FSAC = .000014	WORD15 = .000036	WORD46 = .000134	WORD76 = .000230
S\$LA = .000001	T\$FSB2 = .000010	WORD16 = .000040	WORD47 = .000136	WORD77 = .000232
S\$QB = .000005	T\$IB = .000026	WORD17 = .000042	WORD48 = .000140	WORD78 = .000234
S\$QR = .000006	T\$IBAR = .000024	WORD18 = .000044	WORD49 = .000142	WORD79 = .000236
S\$QX = .000004	T\$IBE = .020000	WORD19 = .000046	WORDS = .000012	WORD8 = .000020
S\$SR = .000007	T\$IBF = .040000	WORD20 = .000050	WORD50 = .000144	WORD80 = .000240
S\$S1 = .000010	T\$ICD = .000040	WORD21 = .000052	WORD51 = .000146	WORD81 = .000242
S\$S2 = .000014	T\$MODE = .004000	WORD22 = .000054	WORD52 = .000150	WORD82 = .000244
TD\$CTR = .176370	T\$OB = .000036	WORD23 = .000056	WORD53 = .000152	WORD83 = .000246
TD\$CTW = .176360	T\$OBE = .004000	WORD24 = .000060	WORD54 = .000154	WORD84 = .000250
TD\$INL = .004000	T\$OBF = .010000	WORD25 = .000062	WORD55 = .000156	WORD85 = .000252
TD\$MEM = .000270	T\$OBRA = .000034	WORD26 = .000064	WORD56 = .000160	WORD86 = .000254
TD\$OAR = .176344	T\$OBWA = .000032	WORD27 = .000066	WORD57 = .000162	WORD87 = .000256
TD\$OTR = .176346	T\$OUTA = .100000	WORD28 = .000070	WORD58 = .000164	WORD88 = .000260
TD\$ORD = .000274	T\$RBD = .000200	WORD29 = .000072	WORD59 = .000166	WORD89 = .000262
TD\$SW = .176376	T\$RNB = .000040	WORD30 = .000074	WORD6 = .000014	WORD9 = .000022
TD\$TAR = .176372	T\$RSET = .040000	WORD31 = .000076	WORD60 = .000170	WORD90 = .000264
TD\$TAW = .176362	T\$SC = .000022	WORD32 = .000080	WORD61 = .000172	WORD91 = .000266
TD\$TDR = .176374	T\$SCLK = .020000	WORD33 = .000082	WORD62 = .000174	WORD92 = .000270
TD\$TDW = .176364	T\$SEG1 = .000000	WORD34 = .000084	WORD63 = .000176	WORD93 = .000272
T\$AD = .000020	T\$SEG2 = .000001	WORD35 = .000086	WORD64 = .000200	WORD94 = .000274
T\$BA = .000002	T\$SEG3 = .000002	WORD36 = .000090	WORD65 = .000202	WORD95 = .000276
T\$BD = .000010	T\$SO = .000001	WORD37 = .000092	WORD66 = .000204	WORD96 = .000300
T\$BSO = .100000	T\$UBUS = .100000	WORD38 = .000094	WORD67 = .000206	WORD97 = .000302
T\$BT = .000020	T\$ICLK = .000400	WORD39 = .000096	WORD68 = .000210	WORD98 = .000304
T\$BTAR = .000030	T\$BBEN = .000020	WORD40 = .000098	WORD69 = .000212	WORD99 = .000306
T\$BTD = .000200	UBD, IN = .000020	WORD41 = .000100	WORD7 = .000016	WORDVAL = .000310
T\$CD = .000100	WORD0 = .000000	WORD42 = .000102	WORD70 = .000214	XTREAD = .001000
T\$CLK = .000200	WORD1 = .000002	WORD43 = .000104	WORD71 = .000216	XTWRT = .000400
T\$DISK = .000200	WORD10 = .000024			

. ABS. 000000 000
000000 001
PPLD 002444 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3729 WORDS (15 PAGES)
DYNAMIC MEMORY: 4916 WORDS (18 PAGES)
ELAPSED TIME: 00:00:56
PPLD, PPLD-SP=[20, 1]IM, [20, 1]PPLD

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/17 : CIA-RDP85-00514R000200020001-3

```

58 000050 000000 QXCNT: .WORD 0 ;NUMBER OF QEX ADDR/DATA PAIRS
59 000052 000000 CODE: .WORD 0 ;CODE FOR MEMORY SELECTION
60 ;
61 ;
62 ;
63 ; PRINT
64 ; PERFORM THIRD LEVEL PARSING
65 ; EG. IN THE COMMAND
66 ; PP>PR.QW.0
67 ; PARSE THE 'QW'
68 ;
69 000054 PPPR4:
70 000054 CALL FIND ;LOCATE MEMORY MNEMONIC IN COMMAND LINE
71 000060 103004 BCC 1$ ;OK, CONTINUE
72 000062 CALL ERR4 ;MISSING OPERAND
73 000066 000167 002066 JMP PR4X ;EXIT
74 ;
75 000072 012700 000044 1$: MOV #PR4LN,R0 ;NUMBER OF TABLE ENTRIES
76 000076 012702 000000 MOV #PR4TBL,R2 ;POINT TO TABLE
77 000102 CALL SCAN ;MATCH AGAINST COMMAND LINE
78 000106 103004 BCC 2$ ;MATCH WAS MADE
79 000110 CALL ERR7 ;INVALID MEMORY MNEMONIC
80 000114 000167 002040 JMP PR4X
81 ;
82 ;
83 ; SAVE POINTER TO ROUTINE ASSOCIATED WITH THE MEMORY MNEMONIC
84 ; SCAN THE REMAINDER OF THE COMMAND LINE FOR CONTROL INFORMATION
85 ;
86 ;
87 ; 1. START ADDRESS ONLY. PRINT ONE MEMORY LOCATION.
88 ; PP>PR.QW.0
89 ;
90 ; 2. START ADDRESS, LOOP INDICATOR, LOOP ON THE READING OF
91 ; THIS ONE MEMORY LOCATION ONLY. PRINT THE CONTENTS ONLY
92 ; ONCE.
93 ; PP>LD.QW.0 L
94 ;
95 ; 3. START ADDRESS, END ADDRESS, NO LOOP. PRINT THE CONTENTS
96 ; OF MEMORY BETWEEN THE START AND END ADDRESSES.
97 ; PP>PR.QW.0 ?
98 ;
99 ; 4. START ADDRESS, END ADDRESS, LOOP INDICATOR, READ THE CONTENTS
100 ; OF MEMORY BETWEEN THE START AND END ADDRESSES, HOWEVER, PRINT
101 ; ONLY THE CONTENTS OF MEMORY AT THE START ADDRESS.
102 ; PP>PR.QW.0 ? L
103 ;
104 000120 010167 000000G 2$: MOV R1,RTNPT ;SAVE POINTER
105 000124 CALL FIND ;LOCATE NON-BLANK IN COMMAND LINE
106 000130 103004 BCC 3$ ;OK, CONTINUE
107 000132 CALL ERR4 ;MISSING OPERAND
108 000136 000167 002016 JMP PR4X ;EXIT
109 000142 3$: CALL PACK ;CONVERT COMMAND LINE VALUE TO BINARY
110 000146 103004 BCC 4$ ;CONVERSION SUCCESSFUL
111 000150 CALL ERR5 ;INVALID NUMERIC VALUE
112 000154 000167 002000 JMP PR4X
113 ;
114 000160 016767 000000G 4$: MOV BINWD,MSTR1 ;SAVE PRINT START ADDRESS
115 000166 016767 000000G MOV BINWD,MSTR2 ;SAVE IT TWICE
116 000174 012767 177777 000000G MOV #1,MEND ;INIT END ADDRESS
117 ;
118 ; START ADDRESS HAS BEEN FOUND. SCAN FOR END ADDRESS OR
119 ; LOOP INDICATOR (CONDITIONS 2, 3 ABOVE). IF THERE IS NOTHING

```


Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

115          ; FURTHER IN THE COMMAND LINE, CONDITION 1 IS IN EFFECT.
116          ;
117 000202.      CALL    FIND          ;SCAN COMMAND LINE.
118 000205. 103004 BCC     5$          ;SOMETHING THERE.
119 000210. 016767 000000G-000000G. MOV    MSTR2,MEND.  ;SET END ADDR. = START ADDR.
120 000216. 000445 BR      9$          ;JUMP TO RTN.
121          ;
122 000220. 122711 000114 5$: CMPB    #1L,(R1)      ;LOOP INDICATOR.
123 000224. 001011 BNE     6$          ;NO, MUST BE UPPER ADDRESS.
124 000226. 052767 000000G-000000G. BIS    #OUT,BASE.  ;SET FLAG FOR OUTPUT CONTROL.
125 000234. 016767 000000G-000000G. MOV    MSTR2,MEND.  ;SET END ADDR. = START ADDR.
126 000242. CALL    HANG          ;HOW TO STOP LOOP.
127 000246. 000431 BR      9$          ;JUMP TO RTN.
128          ;
129 000250. 6$: CALL    PACK          ;CONVERT UPPER ADDRESS.
130 000254. 103004 BCC     7$          ;OK, CONTINUE.
131 000256. CALL    ERR5          ;INVALID NUMERIC.
132 000262. 000167 001672 JMP     PR4X          ;EXIT.
133          ;
134          ; SAVE END ADDRESS (BINARY)
135          ; CHECK FOR LOOP INDICATOR AFTER END ADDRESS (CONDITION 4)
136          ;
137 000266. 016767 000000G-000000G. 7$: MOV    BINWD,MEND.  ;SET UP END ADDRESS.
138 000274. CALL    FIND          ;CHECK FOR LOOP INDICATOR.
139 000300. 103414 BCS     9$          ;NO LOOP.
140 000302. 122711 000114 CMPB    #1L,(R1)      ;CORRECT LOOP INDICATOR.
141 000306. 001404 BEQ     8$          ;YES.
142 000310. CALL    ERR11         ;
143 000314. 000167 001640 JMP     PR4X          ;NO.
144 000320. 052767 000000G-000000G. 8$: BIS    #OUT,BASE.  ;SET OUTPUT CONTROL.
145 000326. CALL    HANG          ;HOW TO STOP LOOP.
146          ;
147 000332. 016701 000000G. 9$: MOV    RTNPT,R1      ;POINT TO ROUTINE.
148 000336. 000171 000000 JMP     @R1          ;JUMP TO ROUTINE.
149          ;
150          ;
151          ; PRINT FROM QEX WINDOW MEMORY.
152          ; PRINT FROM QEX LOCATION MEMORY.
153          ;
154          ;
155 000342. 012767 000042 177502. PR4QW:: MOV    #0$QW,CODE.  ;SET MEMORY CODE = WINDOW.
156 000350. 000403 BR      PDQX          ;
157 000352. 012767 000043 177472. PR4QL:: MOV    #0$QL,CODE.  ;SET CODE = LOCATION
158          ;
159 000360. 316746 000000G. PDQX: MOV    QXHIGH,-(SP)      ;SUPPLY UPPER MEMORY LIMIT.
160 000364. 016746 000000G. MOV    QXLOW,-(SP)          ;LOWER LIMIT.
161 000370. CALL    BUFSET          ;PREPARE FOR LOAD.
162 000374. 103002. BCC     1$          ;NO ERROR.
163 000376. 000167 001556 JMP     PR4X          ;ERROR.
164          ;
165          ; CALCULATE THE NUMBER OF WORDS TO TRANSFER.
166          ; MSTR2 = START ADDRESS.
167          ; MEND = END ADDRESS.
168          ;
169 000402. 016700 000000G. 1$: MOV    MSTR2,R0      ;LOAD START ADDRESS.
170 000406. 016701 000000G. MOV    MEND,R1          ;LOAD END ADDRESS.
171 000412. 100001 SUB     R0,R1          ;GET NUMBER OF WORDS TO PRINT.

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

172 000414 005201 INC R1 ;+1 (BECAUSE OF ZERO-REL ADDRESS)
173 000416 010167 177426 MOV R1,QXCNT ;SAVE COUNT..
174 ;
175 ;
176 ; WRITE THE FLAG ('0' = READ) AND THE TRANSFER COUNT INTO
177 ; THE FIRST TWO LOCATIONS OF CP DATA MEMORY.
178 000422 005046 CLR -(SP) ;START MICROCODE AT 0
179 000424 SEQCS ;
180 000430 005046 CLR -(SP) ;RESET BR INHIBIT.
181 000432 CPCR ;
182 000436 012746 000377 MOV #377,-(SP) ;SET MRP MICRO ADDRESS = X'FF' (JUMP SELF)
183 000442 SEQMM ;
184 000446 005046 CLR -(SP) ;RESET BR INHIBIT.
185 000450 MRPCR ;
186 000454 012767 001000 176422 MOV #Q$REBK,QR$CR2 ;RE-ARM INTERRUPTS.
187 000462 012767 120000 176422 MOV #<Q$SM+Q$ENOP>,QR$CR2 ;SET SEARCH MODE + ENABLE INTERRUPTS.
188 000470 012746 000360 MOV #Q$CSEL,-(SP) ;CLEAR ALL SELECTIONS.
189 000474 052716 001001 BIS #<Q$LBD+Q$LBP>,(SP) ;CLEAR DRIVE AND PULSE.
190 000500 052716 030000 BIS #Q$CNC,(SP) ;CLEAR CP NO-CLOCK.
191 000504 005046 CLR -(SP) ;SET NOTHING.
192 000506 CSRI ;
193 ;
194 000512 005067 177326 CLR CDADD ;WRITE FLAG TO CD ADDR 0
195 000516 005067 177324 CLR CDDAT ;FLAG FOR READ DEX
196 000522 LOADCD ;
197 000526 016767 177316 177312 MOV QXCNT,CDDAT ;WRITE DEX ADDR/DATA PAIRS COUNT TO CD LOC 1
198 000534 LOADCD ;
199 ;
200 ; NOW FILL EVERY OTHER LOCATION IN CP DATA MEMORY
201 ; WITH A DEX ADDRESS (UP TO THE UPPER LIMIT OF
202 ; DEX MEMORY AS ENTERED IN THE COMMAND LINE).
203 ;
204 000540 016767 000000G 177300 2$ MOV MSTR2,CDDAT ;PUT DEX ADDRESS IN CP DATA MEMORY.
205 000546 LOADCD ;LOADCD INCREMENTS CDADD.
206 000552 005267 177266 INC CDADD ;INCREMENT AGAIN - PASS 1 ADDRESS.
207 000556 066767 000000G 000000G ADD INCVAL,MSTR2 ;ADVANCE DEX ADDRESS
208 000564 026767 000000G 000000G CMP MSTR2,MEND ;HAS UPPER MEM LIMIT BEEN REACHED.
209 000572 101762 BLOS 2$ ;NO, CONTINUE.
210 ;
211 000574 005046 CLR -(SP) ;CLEAR NOTHING IN CSRI
212 000576 012746 176000 MOV #Q$CLK,-(SP) ;SET NO-CLOCKS.
213 000602 CSRI ;
214 000606 005067 176422 CLR QR$CR2 ;SET LOAD MODE.
215 ;
216 ;
217 ; RUN THE DEX MICROCODE. MICROCODE READS DEX CONTENTS INTO
218 ; CP DATA MEMORY SLOTS SKIPPED ABOVE. IE, THE DATA IS READ
219 ; IN FOLLOWING EACH ADDRESS. THE FIRST DATA WORD IS IN CP
220 ; DATA MEMORY LOCATION 3.
221 000612 STRTMC:
222 000612 016746 177234 MOV CODE,-(SP) ;SELECT WINDOW OR LOCATION MEMORY.
223 000616 PPCR ;
224 000622 012746 001760 MOV #1760,-(SP) ;START CP MICROCODE AT X'3F0'
225 000626 SEQCS ;
226 000632 005046 CLR -(SP) ;RESET BR INHIBIT.
227 000634 CPCR ;
228 000640 012746 000377 MOV #377,-(SP) ;SET MRP MICRO ADDRESS = X'FF' (JUMP SELF)

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

229 000644
230 000650 005046
231 000652
232 000656 012767 001000 176422
233 000664 012767 022000 176422
234 000672 012746 000360
235 000676 052716 001001
236 000702 052716 176000
237 000706 005046
238 000710
239
240
241
242 000714
243
244 000726
245
246 000740 005046
247 000742 012746 176000
248 000746
249 000752 005067 176422
250
251
252
253 000756 005046
254 000760
255 000764 005046
256 000766
257 000772 012746 000377
258 000776
259 001002 005046
260 001004
261 001010 012767 001000 176422
262 001016 012767 120000 176422
263 001024 012746 000360
264 001030 052716 001001
265 001034 052716 030000
266 001040 005046
267 001042
268
269 001046 016767 000000 000000
270 001054 012767 000003 176762
271 001062 012767 000005 176424 10$:
272 001070 012767 120100 176422
273 001076 016701 176422 5$:
274 001102 032701 000100
275 001106 001373
276
277 001110 016767 176730 176424
278 001116 012767 120040 176422
279 001124 016701 176422 6$:
280 001130 032701 000040
281 001134 001373
282
283 001136 012767 000001 176424
284 001144 012767 120040 176422
285 001152 016701 176422 7$:

```

```

CALL SEQMM
CLR -(SP)
CALL MRPCR
MOV #Q$REBK,QR$CR2
MOV *(Q$LDP+Q$ENOP),QR$CR2
MOV #Q$CSEL, -(SP)
BIS *(Q$LBD+Q$LBP), (SP)
BIS #Q$NCLK, (SP)
CLR -(SP)
CALL CSR1
;
; WAIT FOR INTERRUPT FROM CP
;
; WT3E$ #EFN.3
;
; CLEF$ #EFN.3
;
;
; CLR -(SP)
; MOV #Q$NCLK, -(SP)
; CALL CSR1
; CLR QR$CR2
; PRINT QEX VALUES FROM CP DATA MEMORY
;
; CLR -(SP)
; CALL SEQCS
; CLR -(SP)
; CALL CPR
; MOV #377, -(SP)
; CALL SEQMM
; CLR -(SP)
; CALL MRPCR
; MOV #Q$REBK,QR$CR2
; MOV *(Q$SM+Q$ENOP),QR$CR2
; MOV #Q$CSEL, -(SP)
; BIS *(Q$LBD+Q$LBP), (SP)
; BIS #Q$CNC, (SP)
; CLR -(SP)
; CALL CSR1
;
; MOV MSTR1,MSTR2
; MOV #3,CDADD
; MOV #Q$RDCD,QR$LBR
; MOV *(Q$ATTN+Q$SM+Q$ENOP),QR$CR2
; MOV QR$CR2,R1
; BIT #Q$ATTN,R1
; BNE 5$
;
; MOV CDADD,QR$LBR
; MOV *(Q$CCCP+Q$SM+Q$ENOP),QR$CR2
; MOV QR$CR2,R1
; BIT #Q$CCCP,R1
; BNE 6$
;
; MOV #1,QR$LBR
; MOV *(Q$CCCP+Q$SM+Q$ENOP),QR$CR2
; MOV QR$CR2,R1
;
; RE-ARM INTERRUPTS
; SET LOAD PPS MODE + ENABLE INTERRUPTS
; CLEAR ALL SELECTIONS
; CLEAR DRIVE AND PULSE
; CLEAR ALL NO-CLOCKS
; SET NOTHING
;
; START MICROCODE AT 0
; RESET BR INHIBIT
; SET MRP MICRO ADDRESS = X'FF' (JUMP SELF)
; RESET BR INHIBIT
; RE-ARM INTERRUPTS
; SET SEARCH MODE + ENABLE INTERRUPTS
; CLEAR ALL SELECTIONS
; CLEAR DRIVE AND PULSE
; CLEAR CP NO-CLOCK
; SET NOTHING
;
; REFRESH START ADDRESS
; START PRINTING FROM CP DATA ADDR 3
; MOVE ATTN CODE TO LOD BUS REG
; SET ATTN CODE READY
; READ CSR2
; ATTN CLEAR
; NO READ AGAIN
;
; CD MEMORY START ADDRESS
; SET CC TO CP
; READ CSR2
; IS CC TO CP CLEAR
; NO READ AGAIN
;
; TRANSFER COUNT TO 1 WORD
; SET CC TO CP
; READ CSR2

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

286 001156 032701 000040      BIT      #Q$CCCP,R1      ; IS CC TO CP CLEAR
287 001162 001373      BNE      7$      ; NO, READ AGAIN
288
289 001164 012767 000000G 176424      MOV      #DATA1,QR$LBR      ; CC MEMORY DATA BUFFER
290 001172 012767 120040 176422      MOV      #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ; SET CC TO CP
291
292
293      ;
294 001200      ;
295      ;
296 001212      ;
297      ;
298      ;
299      ;
300 001224 012767 100400 176422      MOV      #<Q$SM+Q$CHB>,QR$CR2 ; CLEAR INTERRUPT (USE HIT BUFFER INT)
301 001232 012767 101000 176422      MOV      #<Q$SM+Q$REBK>,QR$CR2 ; RE-ARM
302 001240 012767 160000 176422      MOV      #<Q$SM+Q$ENBK+Q$ENOP>,QR$CR2 ; ENABLE
303
304      ;
305      ;
306 001246      ;
307      ;
308 001252 062767 000002 176564      ADD      #2,CDADD      ; BUMP CP DATA MEMORY ADDRESS
309 001260 066767 000000G 000000G      ADD      INCVAL,MSTR2 ; ADVANCE ADDRESS
310 001266 026767 000000G 000000G      CMP      MSTR2,MEND ; HAS UPPER MEMORY LIMIT BEEN REACHED
311 001274 101672      BLOS      10$      ; NO, CONTINUE
312 001276 032767 000000G 000000G      BIT      #LOOP,BASE ; LOOP ON ?
313 001304 001402      BEQ      8$      ; NO, EXIT
314 001306 000167 177300      JMP      STRTMC ; START MICROCODE FOR QEX READ
315
316      ;
317      ;
318 001312 012767 100400 176422      MOV      #<Q$SM+Q$CHB>,QR$CR2 ; CLEAR INTERRUPT (USE HIT BUFFER INT)
319 001320 012767 101000 176422      MOV      #<Q$SM+Q$REBK>,QR$CR2 ; RE-ARM
320 001326 012767 160000 176422      MOV      #<Q$SM+Q$ENBK+Q$ENOP>,QR$CR2 ; ENABLE
321
322 001334 005046      CLR      -(SP) ; CLEAR NOTHING IN CSR1
323 001336 012746 176000      MOV      #Q$NCLK,-(SP) ; SET NO-CLOCKS
324 001342      CALL      CSR1
325 001346 005067 176422      CLR      QR$CR2 ; SET LOAD MODE
326 001352 000167 000602      JMP      PR4X ; EXIT
327
328      ;
329      ;
330      ;
331      ;
332      ;
333      ;
334 001356 012767 000046 176466      PR4FP: MOV      #Q$FP,CODE ; SET MEMORY CODE = POINTER
335 001364 000407      BR      PFAL
336 001366 012767 000045 176456      PR4FC: MOV      #Q$FC,CODE ; SET CODE = COUNTER
337 001374 000403      BR      PFAL
338 001376 012767 000044 176446      PR4FO: MOV      #Q$FO,CODE ; SET CODE = OVERFLOW
339
340 001404 016746 000000G      PFAL: MOV      FAHIGH,-(SP) ; SUPPLY MEMORY UPPER LIMIT
341 001410 316746 000000G      MOV      FALOW,-(SP) ; LOWER LIMIT
342 001414      CALL      BUFSET ; PREPARE FOR LOAD

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
343 001420 103006 BCC 1$ ;OK, CONTINUE.
344 001422 012746 077777 MOV #077777,-(SP) ;VALUE FOR QCL POINTER.
345 001426 CALL STOP ;LOAD QCL POINTER.
346 001432 000167 000522 JMP PR4X
347
348 001436 016746 000000G 1$: MOV MSTR2,-(SP) ;LOAD ADDR INTO QCL POINTER.
349 001442 CALL STOP
350 001446 016746 176400 MOV CODE,-(SP) ;SELECT MEMORY.
351 001452 CALL PPCR ;WRITE SELECTION TO CONTROL REG.
352 001456 CALL PPLB ;DO PPS TO LOD BUS.
353 001462 012667 000000G MOV (SP)+,DATA1 ;GET WORD FROM FAL.
354 001466 CALL PRDATA ;PRINT IT.
355 001472 012746 000040 MOV #0$CLR,-(SP) ;CLEAR PPS.
356 001476 CALL PPCR
357
358 001502 066767 000000G-000000G ADD INCVAL,MSTR2 ;ADVANCE ADDRESS.
359 001510 026767 000000G-000000G CMP MSTR2,MEND ;HAS UPPER MEMORY LIMIT BEEN REACHED.
360 001516 101747 BLOS 1$ ;NO, CONTINUE.
361 001520 032767 000000G-000000G BIT #LOOP,BASE ;LOOP ON?
362 001526 001006 BNE 2$ ;YES, CONTINUE.
363 001530 012746 077777 MOV #077777,-(SP) ;VALUE FOR QCL POINTER.
364 001534 CALL STOP ;LOAD QCL POINTER.
365 001540 000167 000414 JMP PR4X ;EXIT.
366 001544 016767 000000G-000000G-2$: MOV MSTR2,MSTR2 ;INIT START ADDRESS.
367 001552 000731 BR 1$ ;AND REPEAT PRINT.
368
369
370
371
372
373
374 001554 PR4QR: MOV LHHIGH,-(SP) ;SUPPLY MEMORY UPPER LIMIT.
375 001560 016746 000000G MOV LLOW,-(SP) ;LOWER LIMIT.
376 001564 CALL BUFSET ;PREPARE FOR LOAD.
377 001570 103012 BCC 1$ ;OK, CONTINUE.
378 001572 012746 000013 MOV #0$QHLT,-(SP) ;HALT.
379 001576 CALL PPCR
380 001602 012746 002000 MOV #2000,-(SP) ;X'400' = ILLEGAL ADDRESS.
381 001606 CALL LBPP ;SEND TO HP.
382 001612 000167 000342 JMP PR4X
383
384 001616 012746 000053 1$: MOV #0$OLA,-(SP) ;SELECT ADDRESS FOR QLB PAGE.
385 001622 CALL PPCR ;WRITE SELECTION TO CONTROL REG.
386 001626 016746 000000G MOV MSTR2,-(SP) ;LOAD ADDR INTO QCL POINTER.
387 001632 CALL LBPP
388 001636 012746 000001 MOV #0$QLR,-(SP) ;SELECT QLB REF PAGE.
389 001642 CALL PPCR
390 001646 CALL PPLB ;DO PPS TO LOD BUS.
391 001652 012667 000000G MOV (SP)+,DATA1 ;GET WORD FROM FAL.
392 001656 CALL PRDATA ;PRINT IT.
393
394 001662 066767 000000G-000000G ADD INCVAL,MSTR2 ;ADVANCE ADDRESS.
395 001670 026767 000000G-000000G CMP MSTR2,MEND ;HAS UPPER MEMORY LIMIT BEEN REACHED.
396 001676 101747 BLOS 1$ ;NO, CONTINUE.
397 001700 032767 000000G-000000G BIT #LOOP,BASE ;LOOP ON?
398 001706 001012 BNE 2$ ;YES, CONTINUE.
399 001710 012746 000013 MOV #0$QHLT,-(SP) ;HALT.
```

```

400 001714          CALL  PPCR
401 001720 012746 002000      MOV  #2000,-(SP)
402 001724          CALL  LBPP
403 001730 000167 000224      JMP  PR4X
404 001734 016767 000000G-000000G-2$: MOV  MSTR1,MSTR2
405 001742 000725          BR    1$
406          ;
407          ;
408          ;
409          ;
410          ;
411          ;
412 001744          PR400::
413 001744 012746 000000      MOV  #0,-(SP)
414 001750 000405          BR    PGSEL2
415 001752          PR401::
416 001752 012746 000001      MOV  #1,-(SP)
417 001756 000402          BR    PGSEL2
418 001760          PR402::
419 001760 012746 000002      MOV  #2,-(SP)
420          ;
421 001764          PGSEL2: CALL  SELPG
422 001770 016746 000000G      MOV  LHHIGH,-(SP)
423 001774 016746 000000G      MOV  LHLOW,-(SP)
424 002000          CALL  BUFSET
425 002004 103012          BCC  1$
426 002006 012746 000013      MOV  #0$0HLT,-(SP)
427 002012          CALL  PPCR
428 002016 012746 002000      MOV  #2000,-(SP)
429 002022          CALL  LBPP
430 002026 000167 000126      JMP  PR4X
431          ;
432 002032 012746 000053      1$: MOV  #0$0LA,-(SP)
433 002036          CALL  PPCR
434 002042 016746 000000G      MOV  MSTR2,-(SP)
435 002046          CALL  LBPP
436 002052 012746 000054      MOV  #0$0LB,-(SP)
437 002056          CALL  PPCR
438 002062          CALL  PPLB
439 002066 012667 000000G      MOV  (SP)+,DATA1
440 002072          CALL  PRDATA
441          ;
442 002076 006767 000000G-000000G      ADD  INCVAL,MSTR2
443 002104 026767 000000G-000000G      CMP  MSTR2,MEND
444 002112 101747          BLOS  1$
445 002114 032767 000000G-000000G      BIT  #LOOP,BASE
446 002122 001012          BNE  2$
447 002124 012746 000013      MOV  #0$0HLT,-(SP)
448 002130          CALL  PPCR
449 002134 012746 002000      MOV  #2000,-(SP)
450 002140          CALL  LBPP
451 002144 000167 000010      JMP  PR4X
452 002150 016767 000000G-000000G-2$: MOV  MSTR1,MSTR2
453 002156 000725          BR    1$
454          ;
455          ;
456 002160          PR4X:

```

PPPR- MACRO-M1110 27-MAR-80 15:34:00 Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

457 002160 012746 000040	MOV	#0\$CLR, -(SP)	: CLEAR PPS
458 002164	CALL	PPCR	
459 002170 042767 000000C 000000G	BIC	*(<ONCE+OUT>, BASE	: CLEAR PRINT CONTROL FLAGS
460 002176	CALL	KILL	: KILL AST (IF THERE WAS ONE)
461 002202 000167 000000G	JMP	PPSXX	

```
463 ;
464 ;
465 ; LOAD CP DATA MEMORY.
466 ;
467 ;
468 002206 LOADCD:
469 002206 012767 000003 176424 MOV. #Q$LD CD,QR$LBR ;MOVE ATTN CODE TO LOD BUS REG
470 002214 012767 120100 176422 MOV. #<Q$ATTN+Q$SM+Q$ENOP>,QR$CR2 ;SET ATTN CODE READY
471 002222 016701 176422 1$: MOV. QR$CR2,R1 ;READ CSR2
472 002226 032701 000100 BIT. #Q$ATTN,R1 ;ATTN CLEAR
473 002232 001373 BNE. 1$ ;NO, READ AGAIN
474 ;
475 002234 016767 175604 176424 MOV. CDADD,QR$LBR ;CD MEMORY START ADDRESS
476 002242 012767 120040 176422 MOV. #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
477 002250 016701 176422 2$: MOV. QR$CR2,R1 ;READ CSR2
478 002254 032701 000040 BIT. #Q$CCCP,R1 ;IS CC TO CP CLEAR
479 002260 001373 BNE. 2$ ;NO, READ AGAIN
480 ;
481 002262 012767 000001 176424 MOV. #1,QR$LBR ;TRANSFER COUNT = 1 WORD
482 002270 012767 120040 176422 MOV. #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
483 002276 016701 176422 3$: MOV. QR$CR2,R1 ;READ CSR2
484 002302 032701 000040 BIT. #Q$CCCP,R1 ;IS CC TO CP CLEAR
485 002306 001373 BNE. 3$ ;NO, READ AGAIN
486 ;
487 002310 012767 000046 176424 MOV. #CDDAT,QR$LBR ;CC MEMORY DATA BUFFER
488 002316 012767 120040 176422 MOV. #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
489 ;
490 ; WAIT FOR INTERRUPT FROM CP.
491 ;
492 002324 WTSE$: #EFN,3
493 ;
494 002336 CLEF$: #EFN,3
495 ;
496 ;
497 ; RE-ARM INTERRUPTS.
498 002350 005267 175470 INC. CDADD ;BUMP CP DATA MEMORY ADDRESS
499 002354 012767 100400 176422 MOV. #<Q$SM+Q$CHB>,QR$CR2 ;CLEAR INTERRUPT (USE HIT BUFFER INT)
500 002362 012767 101000 176422 MOV. #<Q$SM+Q$REBK>,QR$CR2 ;RE-ARM
501 002370 012767 160000 176422 MOV. #<Q$SM+Q$ENBK+Q$ENOP>,QR$CR2 ;ENABLE
502 002376 RETURN
503 ;
504 000001 .END
```


ALUCKE = 040000	BYTE4 = 000004	BYTE91 = 000133	MAROUT = 000002	Q\$CLR = 000040
ALUOE = 004000	BYTE40 = 000050	BYTE92 = 000134	MAR.LO = 002000	Q\$CNC = 030000
A01 = 010000	BYTE41 = 000051	BYTE93 = 000135	MAR.OU = 000040	Q\$CP = 000000
BASE = ***** GX	BYTE42 = 000052	BYTE94 = 000136	MBKALL = 001000	Q\$CPCC = 000010
BINWD = ***** GX	BYTE43 = 000053	BYTE95 = 000137	MBKCLK = 000400	Q\$CP2 = 000200
BITVAL = 000000	BYTE44 = 000054	BYTE96 = 000140	MEND = ***** GX	Q\$CSC = 010000
BIT0 = 000001	BYTE45 = 000055	BYTE97 = 000141	MMADDR = 000100	Q\$CSEL = 000300
3IT1 = 000002	BYTE46 = 000056	BYTE98 = 000142	MMLEFT = 000002	Q\$CSET = 000002
BIT10 = 002000	BYTE47 = 000057	BYTE99 = 000143	MMOE = 000004	Q\$CSP = 020000
BIT11 = 004000	BYTE48 = 000060	BYTVAL = 000144	MMURTE = 000010	Q\$DMA = 000001
BIT12 = 010000	BYTE49 = 000061	CBKALL = 001000	MNOBRE = 100000	Q\$ENBK = 040000
BIT13 = 020000	BYTE5 = 000005	CBKCLK = 000400	MREN1 = 000001	Q\$ENOP = 020000
BIT14 = 040000	BYTE50 = 000062	CDADD = 000044R	MREN2 = 020000	Q\$FAL = 004000
BIT15 = 100000	BYTE51 = 000063	CDAT = 000046R	002.MRPCR = ***** GX	Q\$FC = 000045
BIT2 = 000004	BYTE52 = 000064	CNOBRE = 100000	MSTRT = ***** GX	Q\$FO = 000044
BIT3 = 000010	BYTE53 = 000065	CODE = 000052R	002.MSTR2 = ***** GX	Q\$FP = 000046
BIT4 = 000020	BYTE54 = 000066	CPCCEN = 010000	MSYN = 000040	Q\$HBF = 000002
BIT5 = 000040	BYTE55 = 000067	CPCR = ***** GX	N = 000144	Q\$ICP = 000006
BIT6 = 000100	BYTE56 = 000070	CPREAD = 040000	ONCE = ***** GX	Q\$IH = 000003
BIT7 = 000200	BYTE57 = 000071	CPURTE = 020000	OUT = ***** GX	Q\$IHRL = 000002
BIT8 = 000400	BYTE58 = 000072	CSADDR = 000004	PACK = ***** GX	Q\$IMRP = 000007
BIT9 = 001000	BYTE59 = 000073	CSEQCI = 100000	PFAL = 001404R	002.Q\$LBD = 001000
BUFSET = ***** GX	BYTE6 = 000006	C\$OE = 000040	PGSEL2 = 001764R	002.Q\$LBDP = 001001
BYTE0 = 000000	BYTE60 = 000074	CSR1 = ***** GX	PLB = 000010	Q\$LBP = 000001
BYTE1 = 000001	BYTE61 = 000075	CSURTE = 000100	PLC = 000020	Q\$LCD = 000003
BYTE10 = 000012	BYTE62 = 000076	DATA1 = ***** GX	PLD = 000030	Q\$LDMD = 000004
BYTE11 = 000013	BYTE63 = 000077	DBR.RD = 000001	PLRW = 000200	Q\$LDPP = 002000
BYTE12 = 000014	BYTE64 = 000100	DB\$CPP = 001457	PLR.EN = 000200	Q\$LHP = 010000
BYTE13 = 000015	BYTE65 = 000101	DB\$SPT = 000026	PPCR = ***** GX	Q\$MNC = 140000
BYTE14 = 000016	BYTE66 = 000102	DB\$TPC = 000023	PPLB = ***** GX	Q\$MR = 000052
BYTE15 = 000017	BYTE67 = 000103	DISPGS = 100000	PPR4 = 000054RG	002.Q\$MRP = 000040
BYTE16 = 000020	BYTE68 = 000104	DMAWR = 000005	PPSXX = ***** GX	Q\$MRP2 = 000240
BYTE17 = 000021	BYTE69 = 000105	DMARRD = 000003	POEX = 000360R	002.Q\$MSC = 040000
BYTE18 = 000022	BYTE7 = 000007	DMARWR = 000004	PRDATA = ***** GX	Q\$MSET = 000004
BYTE19 = 000023	BYTE70 = 000106	EFN.3 = ***** GX	PR4FC = 001366RG	002.Q\$MSP = 100000
BYTE2 = 000002	BYTE71 = 000107	ENBR = 010000	PR4FO = 001376RG	002.Q\$NCLK = 170000
BYTE20 = 000024	BYTE72 = 000110	ERR11 = ***** GX	PR4FP = 001356RG	002.Q\$PP = 000100
BYTE21 = 000025	BYTE73 = 000111	ERR4 = ***** GX	PR4LN = 000044 G	Q\$PPSW = 000320
BYTE22 = 000026	BYTE74 = 000112	ERR5 = ***** GX	PR4QL = 000352RG	002.Q\$PP2 = 000300
BYTE23 = 000027	BYTE75 = 000113	ERR7 = ***** GX	PR4QR = 001554RG	002.Q\$QHLT = 000013
BYTE24 = 000030	BYTE76 = 000114	FAHIGH = ***** GX	PR4QW = 000342RG	002.Q\$QL = 000043
BYTE25 = 000031	BYTE77 = 000115	FALOW = ***** GX	PR4Q0 = 001744RG	002.Q\$QLA = 000053
BYTE26 = 000032	BYTE78 = 000116	FIND = ***** GX	PR4Q1 = 001752RG	002.Q\$QLB = 000054
BYTE27 = 000033	BYTE79 = 000117	HANG = ***** GX	PR4Q2 = 001760RG	002.Q\$QLR = 000001
BYTE28 = 000034	BYTE8 = 000010	INCVL = ***** GX	PR4TBL = 000000RG	002.Q\$QW = 000042
BYTE29 = 000035	BYTE80 = 000120	KILL = ***** GX	PR4X = 002160R	002.Q\$RDC = 000005
BYTE3 = 000003	BYTE81 = 000121	LBPP = ***** GX	Q\$SCR1 = 176420	Q\$RDMD = 000006
BYTE30 = 000036	BYTE82 = 000122	LHHIGH = ***** GX	Q\$SCR2 = 176422	Q\$REBK = 001000
BYTE31 = 000037	BYTE83 = 000123	LHLOW = ***** GX	Q\$SLBR = 176424	Q\$RNC = 000000
BYTE32 = 000040	BYTE84 = 000124	LCADCD = 002206R	002.QXCNT = 000050R	002.Q\$RSC = 004000
BYTE33 = 000041	BYTE85 = 000125	LOC.EN = 000100	QXHIGH = ***** GX	Q\$RSET = 000010
BYTE34 = 000042	BYTE86 = 000126	LOC.WA = 040000	QXLOW = ***** GX	Q\$SN = 100000
BYTE35 = 000043	BYTE87 = 000127	LOC.WB = 100000	Q\$ATTN = 000100	Q\$SP = 000120
BYTE36 = 000044	BYTE88 = 000130	LOOP = ***** GX	Q\$BCL = 000001	Q\$SP2 = 000340
BYTE37 = 000045	BYTE89 = 000131	MAREN1 = 000001	Q\$CCCP = 000040	RGQ.EN = 000200
BYTE38 = 000046	BYTE9 = 000011	MAREN2 = 004000	Q\$CHB = 000400	RGQ.VA = 020000
BYTE39 = 000047	BYTE90 = 000132	MARLOD = 010000	Q\$CHRL = 000200	RTNPT = ***** GX

PPPR: M1110 27-MAR-80 15:24 PAGE 6-2
SYMBOL TABLE

Approved For Release 2005/07/11 : CIA-RDP85-00514R000200020001-3

SCAN = ***** GX	T\$CLK = 000200	WORD1 = 000002	WORD40 = 000120	WORD71 = 000216
SELPG = ***** GX	T\$DISK = 000200	WORD10 = 000024	WORD41 = 000122	WORD72 = 000220
SEQCS = ***** GX	T\$DRD = 000004	WORD11 = 000026	WORD42 = 000124	WORD73 = 000222
SEQMM = ***** GX	T\$EMEM = 010000	WORD12 = 000030	WORD43 = 000126	WORD74 = 000224
SEQCI = 000010	T\$FSAA = 000000	WORD13 = 000032	WORD44 = 000130	WORD75 = 000226
STQP = ***** GX	T\$FSAB = 000004	WORD14 = 000034	WORD45 = 000132	WORD76 = 000230
STRMC = 000612R	T\$FSAC = 000014	WORD15 = 000036	WORD46 = 000134	WORD77 = 000232
S\$CLR = 000000	T\$FSB2 = 000010	WORD16 = 000040	WORD47 = 000136	WORD78 = 000234
S\$LA = 000001	T\$IB = 000026	WORD17 = 000042	WORD48 = 000140	WORD79 = 000236
S\$QB = 000005	T\$IBAR = 000024	WORD18 = 000044	WORD49 = 000142	WORD8 = 000020
S\$QR = 000006	T\$IBE = 020000	WORD19 = 000046	WORD5 = 000012	WORD80 = 000240
S\$QX = 000004	T\$IBF = 040000	WORD2 = 000004	WORD50 = 000144	WORD81 = 000242
S\$SR = 000007	T\$ICD = 000040	WORD20 = 000050	WORD51 = 000146	WORD82 = 000244
S\$S1 = 000010	T\$MODE = 004000	WORD21 = 000052	WORD52 = 000150	WORD83 = 000246
S\$S2 = 000014	T\$OB = 000036	WORD22 = 000054	WORD53 = 000152	WORD84 = 000250
TD\$CTR = 176370	T\$OBE = 004000	WORD23 = 000056	WORD54 = 000154	WORD85 = 000252
TD\$CTW = 176360	T\$OBF = 010000	WORD24 = 000060	WORD55 = 000156	WORD86 = 000254
TD\$INL = 004000	T\$OBRA = 000034	WORD25 = 000062	WORD56 = 000160	WORD87 = 000256
TD\$MEM = 000270	T\$OBWA = 000032	WORD26 = 000064	WORD57 = 000162	WORD88 = 000260
TD\$OAR = 176344	T\$OUTA = 100000	WORD27 = 000066	WORD58 = 000164	WORD89 = 000262
TD\$OTR = 176346	T\$RBD0 = 000200	WORD28 = 000070	WORD59 = 000166	WORD9 = 000022
TD\$ORD = 000274	T\$RNB = 000040	WORD29 = 000072	WORD6 = 000014	WORD90 = 000264
TD\$SW = 176376	T\$RSET = 040000	WORD3 = 000006	WORD60 = 000170	WORD91 = 000266
TD\$TAR = 176372	T\$SC = 000022	WORD30 = 000074	WORD61 = 000172	WORD92 = 000270
TD\$TAW = 176362	T\$SCLK = 020000	WORD31 = 000076	WORD62 = 000174	WORD93 = 000272
TD\$TDR = 176374	T\$SEG1 = 000000	WORD32 = 000100	WORD63 = 000176	WORD94 = 000274
TD\$TDW = 176364	T\$SEG2 = 000001	WORD33 = 000102	WORD64 = 000200	WORD95 = 000276
T\$AD = 000020	T\$SEG3 = 000002	WORD34 = 000104	WORD65 = 000202	WORD96 = 000300
T\$BA = 000002	T\$S0 = 000001	WORD35 = 000106	WORD66 = 000204	WORD97 = 000302
T\$BD = 000010	T\$UBUS = 100000	WORD36 = 000110	WORD67 = 000206	WORD98 = 000304
T\$BS0 = 100000	T\$1CLK = 000400	WORD37 = 000112	WORD68 = 000210	WORD99 = 000306
T\$BT = 000020	T\$BBEN = 000020	WORD38 = 000114	WORD69 = 000212	WORDVAL = 000310
T\$BTAR = 000030	UBDIN = 000020	WORD39 = 000116	WORD7 = 000016	XTREAD = 001000
T\$BTD = 002000	WORD0 = 000000	WORD4 = 000010	WORD70 = 000214	XTURTE = 000400
T\$CD = 000100				

. ABS. 000000 000
000000 001
PPPR 002400 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3651 WORDS (15 PAGES)
DYNAMIC MEMORY: 4916 WORDS (18 PAGES)
ELAPSED TIME: 00:00:56
PPPR,PPPR/SP=C20.111M,C20.13PPPR

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3


```

58      ;
59      ;
60      ; STORE
61      ; PERFORM THIRD LEVEL PARSING
62      ; EG. IN THE COMMAND
63      ; PP>ST-0 CR
64      ; PARSE THE '0'
65      ;
66      000050      ST4::
67      000050      CALL    FIND          ;FIND A OPERAND IN COMMAND LINE
68      000054      103004      BCC     1$          ;OK, CONTINUE
69      000056      CALL    ERR4          ;NOTHING THERE
70      000062      000167      000176      JMP     ST4X      ;RETURN TO TOP OF LOOP (PROMPT)
71      000066      1$:      CALL    PACK          ;CONVERT VALUE IN COMMAND LINE TO BINARY
72      000072      103004      BCC     2$
73      000074      CALL    ERR5
74      000100      000167      000160      JMP     ST4X
75      ;
76      ;
77      ; CONTINUE PARSING
78      ; FIND THE REGISTER MNEMONIC IN THE COMMAND LINE
79      000104      2$:      CALL    FIND          ;LOCATE A NON-BLANK
80      000110      103004      BCC     3$
81      000112      CALL    ERR4
82      000116      000167      000142      JMP     ST4X
83      ;
84      ;
85      ; MATCH THE MNEMONIC FROM THE COMMAND LINE AGAINST THE
86      ; TABLE OF VALID MNEMONICS
87      000122      012700      000004      3$:      MOV     #ST4LN,R0      ;NUMBER OF TABLE ENTRIES
88      000126      012702      000000      MOV     #ST4TBL,R2      ;R2 -> TABLE
89      000132      CALL    SCAN          ;LOOK FOR REG MNEMONIC
90      000136      103004      BCC     4$          ;OK, CONTINUE
91      000140      CALL    ERR6
92      000144      000167      000114      JMP     ST4X
93      ;
94      ;
95      ; SAVE THE POINTER TO THE ROUTINE ASSOCIATED WITH THE
96      ; REGISTER, R1 -> ROUTINE ADDRESS
97      ; CALL ROUTINE TO SCAN COMMAND LINE FOR LOOP INDICATOR
98      ; EG. PP>ST-0 CR:L
99      ; LOOP FLAG WILL BE SET IF INDICATOR IS PRESENT
100     ; JUMP TO ROUTINE TO LOAD REGISTER
101     000150      010167      000000G      4$:      MOV     R1,RTNPT      ;SAVE POINTER TO RTN
102     000154      CALL    LOOPR          ;LOOP?
103     000160      016701      000000G      ST4IN:      MOV     RTNPT,R1      ;POINT TO ROUTINE
104     000164      000171      000000      JMP     @<R1>      ;EXECUTE ROUTINE
105     ;
106     ;
107     ; PPS CONTROL REGISTER
108     ;
109     000170      ST4CR::
110     000170      016746      000000G      MOV     BINWD,-(SP)      ;DATA FOR LOD BUS REG
111     000174      CALL    PPCR          ;WRITE TO CONTROL REG
112     000200      000423      BR      ST4LP      ;TEST FOR LOOP
113     ;
114     ; OCL POINTER

```

```
115 ;
116 000202 ST4QP:: MOV. BINWD, -(SP) ;WORD FOR QCL POINTER
117 000202 016746 000000G CALL. STQP ;LOAD QCL POINTER
118 000206 BR ST4LP ;TEST FOR LOOP
119 000212 000416 ;
120 ;
121 ; LOAD MASK REGISTER
122 ;
123 000214 ST4MR:: MOV. #0$MR, -(SP) ;MASK REGISTER SELECT FOR PP CR
124 000214 012746 000052 CALL. PPCR ;LOAD PP CONTROL REG
125 000220 MOV. BINWD, -(SP) ;WORD FOR MASK REGISTER
126 000224 016746 000000G CALL. LBPSC ;SEND MASK WORD
127 000230 BR ST4LP ;TEST FOR LOOP
128 000234 000405 ;
129 ;
130 ; LOAD PPS Q-REG
131 ;
132 000236 ST4QR:: MOV. BINWD, -(SP) ;WORD FOR Q-REG
133 000236 016746 000000G CALL. QREG ;STORE IT
134 000242 BR ST4LP ;TEST FOR LOOP
135 000246 000400 ;
136 ;
137 ;
138 000250 ST4LP: BIT. #LOOP, BASE ;REPEAT?
139 000250 032767 000000G 000000G BEQ. ST4X ;NO, EXIT
140 000256 001402 JMP. ST4IN
141 000260 000167 177674 ;
142 ;
143 000264 ST4X: CALL. KILL
144 000264 JMP. PPSXX
145 000270 000167 000000G
```

```

147      ;
148      ;
149      ;
150      ; READ
151      ; PERFORM THIRD LEVEL PARSING
152      ; EG. IN THE COMMAND:
153      ; PP>RE CR
154      ; PARSE THE 'CR'
155      ;
156      000274      RE4::
157      000274      CALL FIND                ;LOCATE A REG. MNEMONIC
158      000300      103004      BCC 1$
159      000302      CALL ERR4
160      000306      000167 000164      JMP RE4X
161      ;
162      ;
163      ; MATCH THE REGISTER MNEMONIC FROM THE COMMAND LINE AGAINST
164      ; THE TABLE OF VALID MNEMONICS
165      000312      012700 000001      1$: MOV #RE4LN,R0      ;NUMBER OF TABLE ENTRIES
166      000316      012702 000020      MOV #RE4TBL,R2      ;R2-> TABLE
167      000322      CALL SCAN                ;LOOK FOR REG. MNEMONIC
168      000326      103004      BCC 2$      ;OK, CONTINUE
169      000330      CALL ERR6
170      000334      000167 000136      JMP RE4X
171      ;
172      ;
173      ; SAVE THE POINTER TO THE ROUTINE ASSOCIATED WITH THE
174      ; REGISTER, R1 -> ROUTINE ADDRESS
175      ; CALL ROUTINE TO SCAN COMMAND LINE FOR LOOP INDICATOR
176      ; EG. PP>RE CR L
177      ; LOOP FLAG WILL BE SET IF INDICATOR IS PRESENT
178      ; JUMP TO ROUTINE TO READ REGISTER
179      000340      010167 000000G      2$: MOV R1,RTNPT      ;SAVE POINTER TO RTN
180      000344      CALL LOOPR                ;LOOP?
181      000350      016701 000000G      RE4IN: MOV RTNPT,R1      ;POINT TO ROUTINE
182      000354      000171 000000      JMP @R1      ;EXECUTE ROUTINE
183      ;
184      ;
185      ;
186      ; STATUS WORD
187      000360      RE4SW::
188      000360      012746 001001      MOV #Q$LBD+Q$LBP>,-(SP)      ;CLEAR DRIVE AND PULSE
189      000364      052716 000360      BIS #Q$CSEL,(SP)      ;CLEAR SELECT BITS
190      000370      012746 176000      MOV #Q$NCLK,-(SP)      ;SET NO-CLOCKS
191      000374      052716 000320      BIS #Q$PPSW,(SP)      ;READ PPS STATUS WORD
192      000400      CALL CSR1
193      000404      016701 176424      MOV OR$LBR,R1      ;READ LOD BUS REG
194      000410      042701 177774      BIC #177774,R1      ;CLEAR ALL BITS EXCEPT LOW 3
195      ;
196      000414      012746 000320      MOV #Q$PPSW,-(SP)      ;CLEAR PP STATUS REQUEST
197      000420      005046      CLR -(SP)      ;SET NOTHING
198      000422      CALL CSR1
199      000426      000400      BR RE4PUT      ;PRINT AND TEST LOOP FLAG
200      ;
201      ;
202      000430      RE4PUT:
203      000430      032767 000000G 000000G      BIT #ONCE,BASE      ;PRINTED ONCE

```

PPREST- M 00-M1110 27-MAR-80 15:25 PAGE 6-1

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

204 000436 001011		BNE	1\$:YES, SKIP PRINT.
205 000440 052767 000000G.000000G.		BIS	#ONCE,BASE.	:SET FLAG FOR PRINTED ONCE.
206 000446 012705 000000G.		MOV	#PRINT,R5	:POINT TO PRINT LINE
207 000452		CALL	UNPK	:CONVERT VALUE IN R1 FOR PRINTING.
208 000456		CALL	CONSOL.	:PRINT IT.
209				
210 000462 032767 000000G.000000G. 1\$:		BIT	#LOOP,BASE.	:REPEAT ?
211 000470 001402		BEQ	RE4X	:NO, EXIT.
212 000472 000167 177652		JMP	RE4IN.	
213				
214 000476				
215 000476 042767 000000G.000000G.	RE4X:	BIC	#ONCE,BASE.	:CLEAR PRINT CONTROL FLAG.
216 000504		CALL	KILL	:KILL AST.
217 000510 000167 000000G.		JMP	PPSXX.	

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

219      ;
220      ;
221      ;
222      ;
223      000514      ERASE
224      000514      ER4::
225      000520      103004      CALL    FIND      ;FIND A NON-BLANK IN COMMAND LINE
226      000522      ;          BCC     1$      ;OK, CONTINUE
227      000526      000167 000524      CALL    ERR4      ;NOTHING THERE
228      ;          JMP     ER4X      ;RETURN TO TOP OF LOOP (PROMPT)
229      000532      012700 000005      1$:      MOV     #ER4LN,R0      ;NUMBER OF TABLE ENTRIES
230      000536      012702 000024      MOV     #ER4TBL,R2      ;R2 -> TABLE
231      000542      ;          CALL    SCAN      ;LOOK FOR MNEMONIC
232      000546      103004      BCC     2$      ;OK, CONTINUE
233      000550      ;          CALL    ERR6
234      000554      000167 000476      JMP     ER4X
235      ;
236      000560      010167 000000G      2$:      MOV     R1,RTNPT      ;SAVE POINTER TO RTN
237      000564      ;          CALL    LOOPR      ;LOOP?
238      000570      016701 000000G      ER4IN:  MOV     RTNPT,R1      ;POINT TO ROUTINE
239      000574      000171 000000      JMP     @R1      ;EXECUTE ROUTINE
240      ;
241      ;
242      ;
243      ;
244      ;
245      000600      ER4FC::
246      000600      012746 001001      MOV     #<Q$LBD+Q$LBP>,-(SP)      ;CLEAR DRIVE AND PULSE
247      000604      052716 000360      BIS     #Q$CSEL,(SP)      ;CLEAR SELECTION BITS
248      000610      012746 176000      MOV     #Q$NCLK,-(SP)      ;SET NO-CLOCKS
249      000614      ;          CALL    CSR1
250      ;
251      ;
252      ;
253      000620      012746 006000      MOV     #Q$RNC,-(SP)      ;CLEAR PPS NO-CLOCKS (START PPS)
254      000624      005046      CLR     -(SP)      ;SET NOTHING
255      000626      ;          CALL    CSR1
256      ;
257      ;
258      ;
259      ;
260      000632      012746 000025      MOV     #025,-(SP)      ;SET PPCR = X'15'
261      000636      ;          CALL    PPCR
262      ;
263      ;
264      ;
265      ;
266      000642      005067 176424      MOV     Q$LBR      ;
267      000646      005046      CLR     -(SP)      ;CLEAR NOTHING
268      000650      012746 001300      MOV     #<Q$PP2+Q$LBD>,-(SP)      ;SELECT PPS AND SET DRIVE
269      000654      ;          CALL    CSR1
270      ;
271      ;
272      ;
273      000660      012767 004000 176422:  MOV     #Q$FAL,Q$CR2      ;SET FAL LED
274      000666      012767 077777 176424:  MOV     #077777,Q$LBR      ;WRITE 'FAL PARK' TO LOD BUS REG
275      000674      005067 176422      CLR     Q$CR2      ;CLEAR

```



```
276 ;
277 ;
278 ; WAIT 2 MS FOR FAL TO FINISH
279 ; ADD = 3.17US
280 ; DEC = 2.65US
281 ; BNE = 1.87US
282 ;
283 000700 012701 000454 MOV #300,R1 ;EXECUTE 300 TIMES
284 000704 005000 CLR R0 ;ADD NOTHING
285 000706 060000 1$: ADD R0,R0
286 000710 005301 DEC R1
287 000712 001375 BNE 1$
288 ;
289 ;
290 000714 012746 001300 MOV #Q$PP2+Q$LBD>,-(SP) ;CLEAR DRIVE AND DESELECT
291 000720 012746 176000 MOV #Q$NCLK,-(SP) ;SET NO-CLOCKS
292 000724 CALL CSR1
293 ;
294 ;
295 ;
296 000730 012746 000040 MOV #Q$CLR,-(SP) ;CODE FOR CLEAR
297 000734 CALL PPCR
298 000740 000167 000312 JMP ER4X
299 ;
300 ;
301 ;
302 ;
303 ;
304 000744 ER4QR: MOV #Q$LBD+Q$LBP>,-(SP) ;CLEAR DRIVE AND PULSE
305 000744 012746 001001 BIS #Q$CSEL,(SP) ;CLEAR SELECTION BITS
306 000750 052716 000360 MOV #Q$NCLK,-(SP) ;SET NO-CLOCKS
307 000754 012746 176000 CALL CSR1
308 000760
309 ;
310 ;
311 ;
312 000764 012746 006000 MOV #Q$RNC,-(SP) ;CLEAR PPS NO-CLOCKS (START PPS)
313 000770 005046 CLR -(SP) ;SET NOTHING
314 000772 CALL CSR1
315 ;
316 ;
317 ;
318 000776 012746 000141 MOV #141,-(SP) ;SET PPCR = X'61'
319 001002 CALL PPCR
320 ;
321 ;
322 ;
323 ;
324 001006 005067 176424 CLR Q$LBR ;SET LBR = 0
325 001012 005046 CLR -(SP) ;CLEAR NOTHING
326 001014 012746 001300 MOV #Q$PP2+Q$LBD>,-(SP) ;SELECT PPS AND SET DRIVE
327 001020 CALL CSR1
328 ;
329 ;
330 ;
331 001024 012767 010000 176422 MOV #Q$LHP,QR$CR2 ;START LHP
332 001032 005067 176422 CLR QR$CR2 ;CLEAR
```

```
333 ;
334 ;
335 ; WAIT 2 MS FOR LHP TO FINISH
336 ; ADD = 3.17US
337 ; DEC = 2.65US
338 ; BNE = 1.87US
339 001036 012701 000454 MOV #300,R1 ;EXECUTE 300 TIMES
340 001042 005000 CLR R0 ;ADD NOTHING
341 001044 060000 1$: ADD R0,R0
342 001046 005301 DEC R1
343 001050 001375 BNE 1$
344 ;
345 ;
346 ; CLEAR LOD BUS DRIVE, DESELECT PPS
347 001052 012746 001300 MOV #Q$PP2+Q$LBD>,-(SP) ;CLEAR DRIVE AND DESELECT
348 001056 012746 176000 MOV #Q$NCLK,-(SP) ;SET NO-CLOCKS
349 001062 CALL CSR1
350 ;
351 ;
352 ; CLEAR PP CR
353 001066 012746 000040 MOV #Q$CLR,-(SP) ;CODE FOR CLEAR
354 001072 CALL PPCR
355 001076 000167 000154 JMP ER4X
356 ;
357 ;
358 ;
359 ;
360 ; CLEAR QLB PAGES
361 ;
362 001102 ER400:: CLR -(SP) ;SELECT PAGE 0
363 001102 005046 BR PGSEL3
364 001104 000405
365 001106 ER401:: MOV #1,-(SP) ;SELECT PAGE 1
366 001106 012746 000001 BR PGSEL3
367 001112 000402
368 001114 ER402:: MOV #2,-(SP) ;SELECT PAGE 2
369 001114 012746 000002
370 ;
371 001120 PGSEL3: CALL SELPG ;SELECT QLB PAGE
372 001124 012746 001001 MOV #Q$LBD+Q$LBP>,-(SP) ;CLEAR DRIVE AND PULSE
373 001130 052716 000360 B13 #Q$CSEL,(SP) ;CLEAR SELECTION BITS
374 001134 012746 176000 MOV #Q$NCLK,-(SP) ;SET NO-CLOCKS
375 001140 CALL CSR1
376 ;
377 ;
378 ; TURN ON PPS CLOCKS
379 001144 012746 006000 MOV #Q$RNC,-(SP) ;CLEAR PPS NO-CLOCKS (START PPS)
380 001150 005046 CLR -(SP) ;SET NOTHING
381 001152 CALL CSR1
382 ;
383 ;
384 ; SET PPS CONTROL REGISTER
385 001156 012746 000155 MOV #155,-(SP) ;SET PPCR = X'6D'
386 001162 CALL PPCR
387 ;
388 ;
389 ; SET LOD BUS DRIVE, SELECT PPS
```

AIDQR.TSK
MRLD.

MEMORY ALLOCATION MAP TKB
27-MAR-80 18.12

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

*** SEGMENT: MRLD.

R/W MEM. LIMITS: 072364 073557 001174 00636.
DISK BLK LIMITS: 000076 000077 000002 00002.

MEMORY ALLOCATION SYNOPSIS:

SECTION	TITLE	IDENT	FILE
. BLK: (RW, I, LCL, REL, CON)	072364	000000	00000.
MRLD: (RW, I, LCL, REL, CON)	072364	001174	00636.
	072364	001174	00636.
\$\$\$ALVC: (RW, D, LCL, REL, CON)	073560	000000	00000.
\$\$\$RTS: (RW, I, GBL, REL, OVR)	071144	000002	00002.
			MRLD.OBJ:1

GLOBAL SYMBOLS:

LD1. 072374-R. LD1LN. 000002 LD1MD. 073010-R. LD1MM. 072646-R. LD1TBL. 072364-R.

AIDOR.TSK:7 MEMORY ALLOCATION MAP TKB
MRPR: 27-MAR-80 18:12

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

*** SEGMENT: MRPR.

R/W-MEM. LIMITS: 072364 073473 001110 00584.
DISK-BLK-LIMITS: 000100 000101 000002-00002.

MEMORY-ALLOCATION-SYNOPSIS:

SECTION.	TITLE.	IDENT.	FILE.
. BLK.: (RW, I, LCL, REL, CON)	072364	000000	00000.
MRPR.: (RW, I, LCL, REL, CON)	072364	001106	00582.
	072364	001106	00582.
MRPR.			MRPR.OBJ:1
\$\$ALVC: (RW, D, LCL, REL, CON)	073472	000000	00000.
\$\$RTS.: (RW, I, GBL, REL, OVR)	071144	000002	00002.

GLOBAL-SYMBOLS:

PR1 072374-R PR1LN 000002 PR1MD 073062-R PR1MM 072662-R PR1TBL 072364-R

AIDQR.TSK MEMORY ALLOCATION MAP TKB
MRREST: 27-MAR-80 18:12

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

*** SEGMENT: MRREST

R/W MEM LIMITS: 072364 073657 001274 00700.
DISK BLK LIMITS: 000102 000103 000002 000002.

MEMORY ALLOCATION SYNOPSIS:

SECTION	TITLE	IDENT	FILE
BLK: (RW, I, LCL, REL, CON)	072364	000000	000000.
MRREST: (RW, I, LCL, REL, CON)	072364	001274	00700.
	072364	001274	00700.
\$\$\$ALVC: (RW, D, LCL, REL, CON)	073660	000000	000000.
\$\$\$RTS: (RW, I, GBL, REL, OVR)	071144	000002	000002.
		MRREST	MRREST.OBJ:1

GLOBAL SYMBOLS:

CL1	073546-R	RE1	073064-R	REIMP	073176-R	ST1	072462-R	ST1LPG	073032-R	ST1PL	072700-R
LOC1P	073556-R	RE1DW	073322-R	RE1SW	073230-R	ST1BK	072614-R	ST1LS	073024-R	ST1PR	072722-R
LOC1S	073562-R	RE1LN	000004	RE1TBL	072442-R	ST1CR	072602-R	ST1LW	073016-R	ST1QR	072744-R
LOC1W	073566-R	RE1MA	073150-R	RS1	073516-R	ST1LN	000011	ST1MA	072636-R	ST1TBL	072364-R

AIDQR.TSK:7 MEMORY ALLOCATION MAP TKB
MRBUG: 27-MAR-80 10:12

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

*** SEGMENT: MRBUG

R/W MEM LIMITS: 072364 073353 000770 00504.
DISK BLK LIMITS: 000104 000104 000001 00001.

MEMORY ALLOCATION SYNOPSIS:

SECTION	TITLE	IDENT	FILE
. BLK: (RW, I, LCL, REL, CON)	072364	000000	00000.
MRBUG: (RW, I, LCL, REL, CON)	072364	000766	00502.
	072364	000766	00502.
\$\$ALVC: (RW, D, LCL, REL, CON)	073352	000000	00000.
\$\$RTS: (RW, I, GBL, REL, OVR)	071144	000002	00002.
	MRBUG		MRBUG.OBJ:1

GLOBAL SYMBOLS:

AT1 072364-R G01 072672-R OF1 072472-R SS1 072516-R

AIDQR:TSK 5 MEMORY ALLOCATION MAP TKB
CP 27-MAR-80

PAGE 8

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

*** SEGMENT: CP ***

R/W MEM. LIMITS: 072014 073517 001504 00836.
DISK BLK. LIMITS: 000105 000106 000002 00002.

MEMORY ALLOCATION SYNOPSIS:

SECTION	TITLE	IDENT	FILE
BLK: (RW, I, LCL, REL, CON)	072014	000000	00000.
CP: (RW, I, LCL, REL, CON)	072014	001312	00714.
	072014	001312	00714.
CP			CP.OBJ:1
\$\$ALVC: (RW, D, LCL, REL, CON)	073326	000170	00120.
\$\$RTS: (RW, I, GBL, REL, OVR)	071144	000002	00002.

GLOBAL SYMBOLS:

AT2	073326-R	CPXX	073224-R	HRL	072544-R	LR2	073336-R	PR2	073506-R	RS2	073446-R
BCL	072112-R	FR2	073406-R	INDEX	072110-R	DF2	073346-R	REREG	073324-R	SS2	073376-R
CL2	073426-R	G02	073416-R	LDREG	073322-R	PB2	073476-R	RE2	073436-R	ST2	073456-R
CP	073144-R	HLB	072144-R	LD2	073466-R	PC2	073356-R	RG2	073366-R		

AIDOR.TSK:7 MEMORY ALLOCATION MAP TKB
CPBUG1 27-MAR-80 18.12

PAGE 10

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

*** SEGMENT: CPBUG1

R/W MEM LIMITS: 073520 074373 000654 00428.
DISK BLK LIMITS: 000107 000107 000001 00001.

MEMORY ALLOCATION SYNOPSIS:

SECTION	TITLE	IDENT	FILE
. BLK:: (RW, I, LCL, REL, CON)	073520	000000	00000.
CPBUG1: (RW, I, LCL, REL, CON)	073520	000654	00428.
	073520	000654	00428.
\$\$ALVC: (RW, D, LCL, REL, CON)	074374	000000	00000.
\$\$RTS: (RW, I, GBL, REL, OVR)	071144	000002	00002.
	CPBUG1		CPBUG1.OBJ:1

GLOBAL SYMBOLS:

AT2 073524-R LR2 073656-R OF2 073632-R PC2 074010-R RG2 073736-R SS2 074072-R

AIDOR.TSK MEMORY ALLOCATION MAP TKB
CPBUG2 27-MAR-80 18:12

Approved For Release 2005/07/14 : CIA-RDP85-00514R000200020001-3

*** SEGMENT: CPBUG2.

R/W MEM LIMITS: 073520 076167 002450 01320.
DISK BLK LIMITS: 000110 000112 000003 00003.

MEMORY ALLOCATION SYNOPSIS:

SECTION	TITLE	IDENT	FILE
. BLK: (RW, I, LCL, REL, CON)	073520	000000	00000.
CPBUG2: (RW, I, LCL, REL, CON)	073520	002450	01320.
	073520	002450	01320.
\$\$ALVC: (RW, D, LCL, REL, CON)	076170	000000	00000.
\$\$RTS: (RW, I, GBL, REL, OVR)	071144	000002	00002.
	CPBUG2		CPBUG2.OBJ:1

GLOBAL SYMBOLS:

FR2 074510-R G02 073656-R

AIDOR.TSK:7 MEMORY ALLOCATION MAP.TKB
CPREST: 27-MAR-80

PAGE 12

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

*** SEGMENT: CPREST

R/W MEM LIMITS: 073520 075327 001610 00904.
DISK BLK LIMITS: 000113 000114 000002 00002.

MEMORY ALLOCATION SYNOPSIS:

SECTION	TITLE	IDENT	FILE
BLK: (RW, I, LCL, REL, CON)	073520	000000	00000.
CPREST: (RW, I, LCL, REL, CON)	073520	001606	00902.
	073520	001606	00902.
CPREST			CPREST.OBJ:1
ALVC: (RW, D, LCL, REL, CON)	075326	000000	00000.
RTS: (RW, I, GBL, REL, OVR)	071144	000002	00002.

GLOBAL SYMBOLS:

CL2	075316-R	RE2RC	075102-R	RE2R4	075142-R	ST2	074000-R	ST2PB	074302-R	ST2RF	074524-R	ST2R7	074564-R
RE2	074664-R	RE2RD	075076-R	RE2R5	075136-R	ST2BK	074134-R	ST2PC	074334-R	ST2R0	074620-R	ST2R8	074560-R
RE2DD	075034-R	RE2RE	075072-R	RE2R6	075132-R	ST2CR	074120-R	ST2PD	074366-R	ST2R1	074614-R	ST2R9	074554-R
RE2LN	000023	RE2RF	075066-R	RE2R7	075126-R	ST2DA	074420-R	ST2RA	074550-R	ST2R2	074610-R	ST2TBL	073520-R
RE2MA	074750-R	RE2R0	075162-R	RE2R8	075122-R	ST2DD	074462-R	ST2RB	074544-R	ST2R3	074604-R		
RE2MP	075000-R	RE2R1	075156-R	RE2R9	075116-R	ST2LN	000031	ST2RC	074540-R	ST2R4	074600-R		
RE2RA	075112-R	RE2R2	075152-R	RE2TBL	073664-R	ST2MA	074160-R	ST2RD	074534-R	ST2R5	074574-R		
RE2RB	075106-R	RE2R3	075146-R	RS2	075266-R	ST2PA	074250-R	ST2RE	074530-R	ST2R6	074570-R		

AIDOR.TSK
CPLD:

MEMORY ALLOCATION MAP TKB
27-MAR-80

PAGE 13

Approved For Release 2005/07/11 : CIA-RDP85-00514R000200020001-3

*** SEGMENT: CPLD ***

R/W-MEM: LIMITS: 073520 075407 001670 00952:
DISK-BLK: LIMITS: 000115 000116 000002-00002:

MEMORY ALLOCATION SYNOPSIS:

SECTION:	TITLE:	IDENT:	FILE:
. BLK: (RW, I, L, L, REL, CON)	073520	000000	00000.
CPLD: (RW, I, L, L, REL, CON)	073520	001670	00952.
	073520	001670	00952.
\$\$ALVC: (RW, D, LCL, REL, CON)	075410	000000	00000.
\$\$RTS: (RW, I, GBL, REL, OVR)	071144	000002	00002.

GLOBAL SYMBOLS:

LD2: 073540-R LD2BL: 074634-R LD2GD: 074212-R LD2GS: 074012-R LD2HL: 074454-R LD2LN: 000004 LD2TBL: 073520-R

AIDOR.TSK:7
CPPR:

MEMORY ALLOCATION MAP TKB
27-MAR-80 18.12

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

*** SEGMENT: CPPR.

R/W MEM. LIMITS: 073520 075303 001564 00884.
DISK BLK LIMITS: 000117 000120 000002 000002.

MEMORY ALLOCATION SYNOPSIS:

SECTION	TITLE	IDENT	FILE
. BLK: (RW, I, LCL, REL, CON)	073520	000000	00000.
CPPR: (RW, I, LCL, REL, CON)	073520	001564	00884.
	073520	001564	00884.
\$\$\$ALVC: (RW, D, LCL, REL, CON)	075304	000000	00000.
\$\$\$RTS: (RW, I, GBL, REL, OVR)	071144	000002	00002.
			CPPR.OBJ:1

GLOBAL SYMBOLS:

PB2: 074764-R PB2HB 075034-R PB2LN 000003 PR2: 073544-R PR2CS 074032-R PR2TBL 073520-R
PB2BL 075220-R PB2HL 075126-R PB2TBL 073530-R PR2CD 074352-R PR2LN 000002.

AIDOR.TSK MEMORY ALLOCATION MAP TKB
BCE 27-MAR-88

PAGE 15

Approved For Release 2005/07/11 : CIA-RDP85-00514R000200020001-3

*** SEGMENT: BCE...

R/W MEM. LIMITS: 072014 072647 000634 00412.
DISK BLK LIMITS: 000121 000121 000001 00001.

MEMORY ALLOCATION SYNOPSIS:

SECTION	TITLE	IDENT	FILE
. BLK: (RW, I, LCL, REL, CON)	072014	000000	00000.
BCE: (RW, I, LCL, REL, CON)	072014	000164	00116.
	072014	000164	00116.
BCREST: (RW, I, LCL, REL, CON)	072200	000446	00294.
	072200	000446	00294.
\$\$\$ALVC: (RW, D, LCL, REL, CON)	072646	000000	00000.
\$\$\$RTS: (RW, I, GBL, REL, OVR)	071144	000002	00002.
	BCE		BCE.OBJ;1
	BCREST		BCREST.OBJ;1

GLOBAL SYMBOLS:

BCE	072030-R	RE3C1	072510-R	RE3LN	000003	ST3	072230-R	ST3LB	072370-R
CEXX	072110-R	RE3C2	072516-R	RE3TBL	072214-R	ST3C1	072350-R	ST3LN	000003
RE3	072424-R	RE3LB	072524-R	RS3	072616-R	ST3C2	072360-R	ST3TBL	072200-R

AIDOR.TSK:7 MEMORY ALLOCATION MAP TKB
PPS: 27-MAR-80

PAGE 16

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

*** SEGMENT: PPS...

R/W MEM LIMITS: 072014 072567 000554 00364.
DISK BLK LIMITS: 000122 000122 000001 00001.

MEMORY ALLOCATION SYNOPSIS:

SECTION	TITLE	IDENT	FILE
. BLK: (RW, I, LCL, REL, CON)	072014	000000	00000.
PPS: (RW, I, LCL, REL, CON)	072014	000464	00308.
	072014	000464	00308.
\$\$ALVC: (RW, D, LCL, REL, CON)	072500	000070	00056.
\$\$RTS: (RW, I, GBL, REL, OVR)	071144	000002	00002.

PPS PPS.OBJ:1

GLOBAL SYMBOLS:

CL4:	072520-R	ER4	072530-R	PPPR4	072510-R	PPSXX	072132-R	QREG	072222-R	STOP	072360-R
CODE:	072050-R	LD4	072500-R	PPS	072052-R	PS4	072540-R	RE4	072550-R	ST4	072560-R

AIDOR.TSK MEMORY ALLOCATION MAP TKB
PPLD. 27-MAR-80

PAGE 12

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

*** SEGMENT: PPLD...

R/W MEM LIMITS: 072570 075233 002444 01316.
DISK BLK LIMITS: 000123 000125 000003 00003.

MEMORY ALLOCATION SYNOPSIS:

SECTION..	TITLE..	IDENT.	FILE..
-----	-----	-----	-----
. BLK: (RW, I, LCL, REL, CON)	072570	000000	00000.
PPLD: (RW, I, LCL, REL, CON)	072570	002444	01316.
	072570	002444	01316.
\$\$\$ALVC: (RW, D, LCL, REL, CON)	075234	000000	00000.
\$\$\$RTS: (RW, I, GBL, REL, OVR)	071144	000002	00002.

GLOBAL SYMBOLS:

LD4. 072640-R LD4FP. 073452-R LD4QL. 073122-R LD4QW. 073112-R LD4Q1. 074246-R LD4TBL. 072570-R
LD4FC. 073462-R LD4LN. 000010 LD4QR. 073754-R LD4Q0. 074240-R LD4Q2. 074254-R

AIDQR.TSK:7 MEMORY ALLOCATION MAP.TKB
PPPR: 27-MAR-80 18:12

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

*** SEGMENT: PPPR.

R/W MEM. LIMITS: 072570 075167 002400 01280.
DISK BLK LIMITS: 000126 000130 000003 00003.

MEMORY ALLOCATION SYNOPSIS:

SECTION.	TITLE.	IDENT.	FILE.
. BLK: (RW, I, LCL, REL, CON)	072570	000000	00000.
PPPR: (RW, I, LCL, REL, CON)	072570	002400	01280.
	072570	002400	01280.
\$\$\$ALVC: (RW, D, LCL, REL, CON)	075170	000000	00000.
\$\$\$RTS: (RW, I, GBL, REL, OVR)	071144	000002	00002.

PPPR: PPPR.OBJ:1

GLOBAL SYMBOLS:

PPPR4 072644-R PR4F0 074166-R PR4LN 000044 PR4QR 074344-R PR4Q0 074534-R PR4Q2 074550-R
PR4FC 074156-R PR4FP 074146-R PR4QL 073142-R PR4QW 073132-R PR4Q1 074542-R PR4TBL 072570-R

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

AIDQR.TSK MEMORY ALLOCATION MAP TKB
PPREST 27-MAR-80

PAGE 19

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

*** SEGMENT: PPREST.

R/W MEM. LIMITS: 072570 074167 001400 00768.
DISK BLK. LIMITS: 000131 000132 000002 00002.

MEMORY ALLOCATION SYNOPSIS:

SECTION	TITLE	IDENT	FILE
BLK: (RW, I, LCL, REL, CON)	072570	000000	00000.
PPREST: (RW, I, LCL, REL, CON)	072570	001376	00766.
	072570	001376	00766.
\$\$\$ALVC: (RW, D, LCL, REL, CON)	074166	000000	00000.
\$\$\$RTS: (RW, I, GBL, REL, OVR)	071144	000002	00002.

GLOBAL SYMBOLS:

CL4	074156-R	ER4LN	000005	ER4Q1	073676-R	PS4	074072-R	RE4SW	073150-R	ST4CR	072760-R	ST4QP	072772-R
ER4	073304-R	ER4QR	073534-R	ER4Q2	073704-R	RE4	073064-R	RE4TBL	072610-R	ST4LN	000004	ST4QR	073026-R
ER4FC	073370-R	ER4Q0	073672-R	ER4TBL	072614-R	RE4LN	000001	ST4	072640-R	ST4MR	073004-R	ST4TBL	072570-R

AIDQR.TSK:7 MEMORY ALLOCATION MAP TKB
SP. 27-MAR-80

PAGE 20

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

*** SEGMENT: SF ***

R/W MEM. LIMITS: 072014 072573 000560 00368.
DISK.BLK.LIMITS: 000133 000133 000001 00001.

MEMORY ALLOCATION SYNOPSIS:

SECTION...	TITLE..	IDENT.	FILE..
. BLK: (RW, I, LCL, REL, CON)	072014	000000	00000.
SP. : (RW, I, LCL, REL, CON)	072014	000172	00122.
	072014	000172	00122.
SPSUB: (RW, I, LCL, REL, CON)	072206	000324	00212.
	072206	000324	00212.
\$\$\$ALVC: (RW, D, LCL, REL, CON)	072532	000040	00032.
\$\$\$RTS: (PW, I, GBL, REL, OVR)	071144	000002	00002.

GLOBAL SYMBOLS:

CL5.	072552-R.	LBSP.	072206-R.	LDS	072532-R.	SPCR.	072450-R.	SPPR5	072542-R.	SPSXX	072116-R
CODE.	072034-R.	LBSSC.	072304-R.	PSS	072562-R.	SPLB.	072402-R.	SPS.	072036-R.		

AIDDR:TSK MEMORY ALLOCATION MAP TKB
SPLD: 27-MAR-80

PAGE 21

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

*** SEGMENT: SPLD ***

R/W MEM LIMITS: 072574 073647 001054 00556.
DISK BLK LIMITS: 000134 000135 000002 00002.

MEMORY ALLOCATION SYNOPSIS:

SECTION...	TITLE..	IDENT.	FILE..
BLK: (RW, I, LCL, REL, CON)	072574	000000	00000.
SPLD: (RW, I, LCL, REL, CON)	072574	001052	00554.
	072574	001052	00554.
\$\$\$ALVC: (RW, D, LCL, REL, CON)	073646	000000	00000.
\$\$\$RTS: (RW, I, GBL, REL, OVR)	071144	000002	00002.
			SPLD.OBJ:1

GLOBAL SYMBOLS:

LD5. 072624-R. LD5QB. 073106-R. LD5QX. 073076-R. LD5S1. 073312-R. LD5TBL. 072574-R.
LD5LN. 000006 LD5QR. 073116-R. LD5SR. 073126-R. LD5S2. 073322-R.

AIDQR.TSK:7 MEMORY ALLOCATION MAP.TKR
SPPR: 27-MAR-88 18:12

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

*** SEGMENT: SPPR.

R/W-MEM. LIMITS: 072574 073547 000754 00492.
DISK-BLK. LIMITS: 000136 000136 000001 00001.

MEMORY-ALLOCATION-SYNOPSIS:

SECTION...	TITLE...	IDENT...	FILE...
. BLK: (RW, I, LCL, REL, CON)	072574	000000	00000.
SPPR: (RW, I, LCL, REL, CON)	072574	000752	00490.
	072574	000752	00490.
\$\$ALVC: (RW, D, LCL, REL, CON)	073546	000000	00000.
\$\$RTS: (RW, I, GBL, REL, OVR)	071144	000002	00002.
			SPPR.OBJ:1

GLOBAL-SYMBOLS:

PR5LN: 000030 PR5OR: 073132-R PR5SR: 073142-R PR5S2: 073274-R SPPR5: 072624-R
PR5QB: 073122-R PR5QX: 073112-R PR5S1: 073264-R PR5TBL: 072574-R

AIDQR.TSK MEMORY ALLOCATION MAP TKB
SPREST 27-MAR-80 19412

PAGE 23

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

*** SEGMENT: SPREST.

R/W MEM. LIMITS: 072574 072613 000020 00016.
DISK BLK. LIMITS: 000137 000137 000001 00001.

MEMORY ALLOCATION SYNOPSIS:

SECTION...	TITLE...	IDENT...	FILE...
. BLK: (RW, I, LCL, REL, CON)	072574	000000	00000.
SPREST: (RW, I, LCL, REL, CON)	072574	000020	00016.
	072574	000020	00016.
\$\$ALVD: (RW, D, LCL, REL, CON)	072614	000000	00000.
\$\$RTS: (RW, I, GBL, REL, OVR)	071144	000002	00002.

GLOBAL SYMBOLS:

CL5 072604-R PS5 072574-R.

*** TASK BUILDER STATISTICS:

TOTAL WORK FILE REFERENCES: 70839.
WORK FILE READS: 0.
WORK FILE WRITES: 0.
SIZE OF CORE POOL: 15082. WORDS (58. PAGES)
SIZE OF WORK FILE: 13568. WORDS (53. PAGES)

ELAPSED TIME: 00:00:54

```

390 001166 005046          CLR:      -(SP)          ;CLEAR NOTHING.
391 001170 012746 001300    MOV:      #<Q$PP2+Q$LBD>,-(SP) ;SELECT PPS AND SET DRIVE.
392 001174          CALL:      CSR1
393          ;
394          ;
395          ;
396 001200 012767 010000 176422 MOV:      #Q$LHP,QR$CR2.    ;START LHP
397 001206 005067 176422    CLR:      QR$CR2.          ;CLEAR
398          ;
399          ;
400          ;
401          ;
402          ;
403          ;
404 001212 012701 000454          MOV:      #300,R1          ;EXECUTE 300 TIMES
405 001216 005000          CLR:      R0              ;ADD NOTHING
406 001220 060000          ADD:      R0,R0
407 001222 005301          DEC:      R1
408 001224 001375          BNE:      1$
409          ;
410          ;
411          ;
412 001226 012746 001300    MOV:      #<Q$PP2+Q$LBD>,-(SP) ;CLEAR DRIVE AND DESELECT
413 001232 012746 176000    MOV:      #Q$NCLK,-(SP)      ;SET NO-CLOCKS
414 001236          CALL:      CSR1
415          ;
416          ;
417          ;
418 001242 012746 000040    MOV:      #Q$CLR,-(SP)          ;CODE FOR CLEAR
419 001246          CALL:      PPCR
420 001252 000167 000000    JMP:      ER4X
421          ;
422          ;
423 001256          ER4X:
424 001256 032767 000000G 000000G BIT:      #LOOP,BASE.    ;LOOP ON ERASE
425 001264 001402          BEQ:      1$              ;NO
426 001266 000167 177276    JMP:      ER4IN.          ;REPEAT ERASE
427          ;
428 001272          1$:
429 001276 000167 000000G    CALL:      KILL          ;KILL AST
                          JMP:      PPSXX

```

```
431 ;
432 ;
433 ; PAGE SWITCH
434 ;
435 ;
436 001302 PS4::
437 001302 CALL FIND
438 001306 103004 BCC 1$
439 001310 CALL ERR4
440 001314 000167 000042 JMP PS4X
441 ;
442 001320 1$: CALL PACK
443 001324 103004 BCC 2$
444 001326 CALL ERR5
445 001332 000167 000024 JMP PS4X
446 ;
447 001336 2$: CALL LOOPR
448 001342 016746 000000G MOV BINWD, -(SP)
449 001346 3$: CALL SELPG
450 001352 032767 000000G 000000G BIT *LOOP.BASE
451 001360 001370 BNE 3$
452 ;
453 ;
454 001362 PS4X:
455 001362 000167 000000G JMP PPSXX
456 ;
457 ;
458 ;
459 ; CALL HQR-LOADER
460 ;
461 ;
462 001366 CL4::
463 001366 CALL CL
464 001372 000167 000000G JMP PPSXX
465 ;
466 000001 .END
```

:FIND-A-NON-BLANK-IN-COMMAND-LINE
:OK-CONTINUE
:NOTHING-THERE
:RETURN-TO-TOP-OF-LOOP-(PROMPT)

:CONVERT-VALUE-IN-COMMAND-LINE-TO-BINARY

:TEST-FOR-LOOP-OPTION
:SELECT-A-QLB-PAGE

:REPEAT-?
:YES

:CALL-ROUTINE-IN-'MAIN'

ALUCKE = 040000	BYTE40 = 000050	BYTE92 = 000134	MAR.LD = 002000	Q\$LBDP = 001001
ALUOE = 004000	BYTE41 = 000051	BYTE93 = 000135	MAR.OU = 000040	Q\$LBP = 000001
A01 = 010000	BYTE42 = 000052	BYTE94 = 000136	MBKALL = 001000	Q\$LDOD = 000003
BASE = ***** GX	BYTE43 = 000053	BYTE95 = 000137	MBKCLK = 000400	Q\$LDMD = 000004
BINWD = ***** GX	BYTE44 = 000054	BYTE96 = 000140	MMADRD = 000100	Q\$LDPP = 002000
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	MMLEFT = 000002	Q\$LHP = 010000
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	MMOE = 000004	Q\$MNC = 140000
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	MMURTE = 000010	Q\$MR = 000052
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	MNOBRE = 100000	Q\$MRP = 000040
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	MREN1 = 000001	Q\$MRP2 = 000240
BIT12 = 010000	BYTE50 = 000062	CBKCLK = 000400	MREN2 = 020000	Q\$MSC = 040000
BIT13 = 020000	BYTE51 = 000063	CL = ***** GX	MSYN = 000040	Q\$MSET = 000004
BIT14 = 040000	BYTE52 = 000064	CL4 = 001366RG	002.N = 000144	Q\$MSP = 100000
BIT15 = 100000	BYTE53 = 000065	CNOBRE = 100000	ONCE = ***** GX	Q\$NCLK = 176000
BIT2 = 000004	BYTE54 = 000066	CONSOL = ***** GX	PACK = ***** GX	Q\$PP = 000100
BIT3 = 000010	BYTE55 = 000067	CPCCEN = 010000	PGSEL3 = 001120R	002.Q\$PPSW = 000320
BIT4 = 000020	BYTE56 = 000070	CPREAD = 040000	PLB = 000010	Q\$PP2 = 000300
BIT5 = 000040	BYTE57 = 000071	CPURTE = 020000	PLC = 000020	Q\$QHLT = 000013
BIT6 = 000100	BYTE58 = 000072	CSADRD = 000004	PLD = 000030	Q\$QL = 000043
BIT7 = 000200	BYTE59 = 000073	CSEQCI = 100000	PLRWR = 000200	Q\$QLA = 000053
BIT8 = 000400	BYTE60 = 000074	CSOE = 000040	PLREN = 000200	Q\$QLB = 000054
BIT9 = 001000	BYTE61 = 000075	CSR1 = ***** GX	PPCR = ***** GX	Q\$QLR = 000001
BYTE0 = 000000	BYTE62 = 000076	CSURTE = 000100	PPSXX = ***** GX	Q\$QW = 000042
BYTE1 = 000001	BYTE63 = 000077	DBR.RD = 000001	PRINT = ***** GX	Q\$RDCD = 000005
BYTE10 = 000012	BYTE64 = 000100	DB\$CPP = 001457	PS4 = 001302RG	002.Q\$RDMD = 000006
BYTE11 = 000013	BYTE65 = 000101	DB\$SPT = 000026	PS4X = 001362R	002.Q\$REBK = 001000
BYTE12 = 000014	BYTE66 = 000102	DB\$TPC = 000023	QREG = ***** GX	Q\$RNC = 006000
BYTE13 = 000015	BYTE67 = 000103	DISPGS = 100000	Q\$CR1 = 176420	Q\$RSC = 004000
BYTE14 = 000016	BYTE68 = 000104	DMAWR = 000005	Q\$CR2 = 176422	Q\$RSET = 000010
BYTE15 = 000017	BYTE69 = 000105	DMARRD = 000003	Q\$LBRE = 176424	Q\$SM = 100000
BYTE16 = 000020	BYTE70 = 000106	DMARWR = 000004	Q\$ATTN = 000100	Q\$SP = 000120
BYTE17 = 000021	BYTE71 = 000107	ENBR = 010000	Q\$BCL = 000001	Q\$SP2 = 000340
BYTE18 = 000022	BYTE72 = 000110	ERR4 = ***** GX	Q\$CCCP = 000040	RE4 = 000274RG
BYTE19 = 000023	BYTE73 = 000111	ERR5 = ***** GX	Q\$CHB = 000400	RE4IN = 000350R
BYTE20 = 000024	BYTE74 = 000112	ERR6 = ***** GX	Q\$CHRL = 000200	RE4LN = 000001 G
BYTE21 = 000025	BYTE75 = 000113	ER4 = 000514RG	002.Q\$CLR = 000040	RE4PUT = 000430R
BYTE22 = 000026	BYTE76 = 000114	ER4FC = 000600RG	002.Q\$CNC = 030000	RE4SW = 000360RG
BYTE23 = 000027	BYTE77 = 000115	ER4IN = 000570R	002.Q\$CP = 000060	RE4TBL = 000020RG
BYTE24 = 000030	BYTE78 = 000116	ER4LN = 000005 G	002.Q\$CPC = 000010	RE4X = 000475R
BYTE25 = 000031	BYTE79 = 000117	ER4QR = 000744RG	002.Q\$CP2 = 000260	RGQ.EN = 000200
BYTE26 = 000032	BYTE80 = 000120	ER4Q0 = 001102RG	002.Q\$CSC = 010000	RGQ.VA = 020000
BYTE27 = 000033	BYTE81 = 000121	ER4Q1 = 001106RG	002.Q\$CSEL = 000360	RGQ.VA = ***** GX
BYTE28 = 000034	BYTE82 = 000122	ER4Q2 = 001114RG	002.Q\$CSET = 000002	SCAN = ***** GX
BYTE29 = 000035	BYTE83 = 000123	ER4TBL = 000024RG	002.Q\$CSP = 020000	SELPG = ***** GX
BYTE30 = 000036	BYTE84 = 000124	ER4X = 001256R	002.Q\$DMA = 000001	SEQ.CI = 000010
BYTE31 = 000037	BYTE85 = 000125	FIND = ***** GX	Q\$ENBK = 040000	STQP = ***** GX
BYTE32 = 000040	BYTE86 = 000126	KILL = ***** GX	Q\$ENOP = 020000	ST4 = 000050RG
BYTE33 = 000041	BYTE87 = 000127	LBPSC = ***** GX	Q\$FAL = 004000	ST4CR = 000170RG
BYTE34 = 000042	BYTE88 = 000130	LOC.EN = 000100	Q\$FC = 000045	ST4IN = 000160R
BYTE35 = 000043	BYTE89 = 000131	LOC.WA = 040000	Q\$FO = 000044	ST4LN = 000004 G
BYTE36 = 000044	BYTE90 = 000132	LOC.WB = 100000	Q\$FP = 000046	ST4LP = 000250R
BYTE37 = 000045	BYTE91 = 000133	LOOP = ***** GX	Q\$HFB = 000002	ST4MR = 000214RG
BYTE38 = 000046		LOOPR = ***** GX	Q\$ICP = 000006	ST4OP = 000202RG
BYTE39 = 000047		MAREN1 = 000001	Q\$IH = 000003	ST4OR = 000236RG
BYTE4 = 000004		MAREN2 = 004000	Q\$IHLR = 000002	ST4TBL = 000000RG
		MARLOD = 010000	Q\$IMRP = 000007	ST4X = 000264R
		MAROUT = 000002	Q\$RBL = 001000	ST4CLR = 000000

PPREST: MACRO-M1110 27-MAR-80 15:25 PAGE 3-2
SYMBOL TABLE:

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

S#LA = 000001	T#FSAC = 000014	WORD13 = 000032	WORD43 = 000126	WORD73 = 000222
S#QB = 000005	T#FSB2 = 000010	WORD14 = 000034	WORD44 = 000130	WORD74 = 000224
S#QR = 000006	T#IB = 000026	WORD15 = 000036	WORD45 = 000132	WORD75 = 000226
S#QX = 000004	T#IBAR = 000024	WORD16 = 000040	WORD46 = 000134	WORD76 = 000230
S#SR = 000007	T#IBE = 020000	WORD17 = 000042	WORD47 = 000136	WORD77 = 000232
S#S1 = 000010	T#IBF = 040000	WORD18 = 000044	WORD48 = 000140	WORD78 = 000234
S#S2 = 000014	T#ICD = 000040	WORD19 = 000046	WORD49 = 000142	WORD79 = 000236
TD#CTR = 176370	T#MODE = 004000	WORD2 = 000004	WORD5 = 000012	WORD8 = 000020
TD#CTW = 176360	T#OB = 000036	WORD20 = 000050	WORD50 = 000144	WORD80 = 000240
TD#INL = 004000	T#OBE = 004000	WORD21 = 000052	WORD51 = 000146	WORD81 = 000242
TD#MEM = 000270	T#OBF = 010000	WORD22 = 000054	WORD52 = 000150	WORD82 = 000244
TD#OHR = 176344	T#OBRA = 000034	WORD23 = 000056	WORD53 = 000152	WORD83 = 000246
TD#OTR = 176346	T#OBWA = 000032	WORD24 = 000060	WORD54 = 000154	WORD84 = 000250
TD#QRD = 000274	T#OUTA = 100000	WORD25 = 000062	WORD55 = 000156	WORD85 = 000252
TD#SW = 176376	T#RBD = 000200	WORD26 = 000064	WORD56 = 000160	WORD86 = 000254
TD#TAR = 176372	T#RNB = 000040	WORD27 = 000066	WORD57 = 000162	WORD87 = 000256
TD#TAW = 176362	T#RSET = 040000	WORD28 = 000070	WORD58 = 000164	WORD88 = 000260
TD#TDR = 176374	T#SC = 000022	WORD29 = 000072	WORD59 = 000166	WORD89 = 000262
TD#TDW = 176364	T#SCLK = 020000	WORD3 = 000006	WORD6 = 000014	WORD9 = 000022
T#AD = 000020	T#SEG1 = 000000	WORD30 = 000074	WORD60 = 000170	WORD90 = 000264
T#BA = 000002	T#SEG2 = 000001	WORD31 = 000076	WORD61 = 000172	WORD91 = 000266
T#BD = 000010	T#SEG3 = 000002	WORD32 = 000100	WORD62 = 000174	WORD92 = 000270
T#BSO = 100000	T#SO = 000001	WORD33 = 000102	WORD63 = 000176	WORD93 = 000272
T#BT = 000020	T#UBUS = 100000	WORD34 = 000104	WORD64 = 000200	WORD94 = 000274
T#BTAR = 000030	T#ICLK = 000400	WORD35 = 000106	WORD65 = 000202	WORD95 = 000276
T#BTD = 002000	T#BBEN = 000020	WORD36 = 000110	WORD66 = 000204	WORD96 = 000300
T#CD = 000100	UBD, IN = 000020	WORD37 = 000112	WORD67 = 000206	WORD97 = 000302
T#CLK = 002000	UNPK = ***** GX	WORD38 = 000114	WORD68 = 000210	WORD98 = 000304
T#DISK = 000200	WORD0 = 000000	WORD39 = 000116	WORD69 = 000212	WORD99 = 000306
T#DRD = 000004	WORD1 = 000002	WORD4 = 000010	WORD7 = 000016	WORDVAL = 000310
T#EMEM = 010000	WORD10 = 000024	WORD40 = 000120	WORD70 = 000214	XTREAD = 001000
T#FSAA = 000000	WORD11 = 000026	WORD41 = 000122	WORD71 = 000216	XTWRITE = 000400
T#FSAB = 000004	WORD12 = 000030	WORD42 = 000124	WORD72 = 000220	

. ABS. 000000 000
000000 001
PPREST: 001376 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3336 WORDS (14 PAGES)
DYNAMIC MEMORY: 4916 WORDS (18 PAGES)
ELAPSED TIME: 00:00:49
PPREST, PPREST/SP=[20,1]IM,[20,1]PPREST:

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

- 1
- 2.
- 3 000000

```
.TITLE.  SP.
.PSECT.  SP.
```

HARDWARE · QUERY · RESOLVER · *MANUAL* DEBUGGING AIDS ·
SUBDOCUMENT · PROCESSORS

PARSE THE COMMAND LINE AND TRANSFER CONTROL TO ONE OF THE
SP SUB-MODULES OR BACK TO QMAIN. CONTROL COULD HAVE BEEN
PASSED TO SP IN ONE OF THREE WAYS:

1. FROM QMAIN. IF SP WAS SELECTED AS THE FIRST PROCESSOR UPON ENTRY TO THE PROGRAM.
 >SP LD QX 0
2. FROM QMAIN. IF SP WAS SELECTED FROM ONE OF QMAIN'S OTHER SUB-MODULES.
 CP>SP LD QX 0
3. ON RETURN FROM ONE OF SP'S SUB-MODULES.
 SP>LD QX 0

```

SP PARSES AT THE SECOND LEVEL OF CONTROL (SEE NOTES AT
QMAIN ON LEVELS OF CONTROL). IN THE EXAMPLES ABOVE, SP
WOULD PARSE 'LD' AND TRANSFER CONTROL TO THE SP SUB-
MODULE SPLD. IF SP SHOULD ENCOUNTER A STRING WHICH IS
NOT A VALID COMMAND MNEMONIC, SP CONSIDERS THE STRING TO
BE A PROCESSOR MNEMONIC AND RETURNS CONTROL TO QMAIN. EG:
SP>CP LD CS 0
THE STRING 'CP' IS NOT A VALID SP (SECOND LEVEL) COMMAND.
SP RETURNS CONTROL TO QMAIN WHICH IN TURN WILL TRANSFER
CONTROL TO ITS SUB-MODULE CP.

```

```
SP SUB-MODULES:
SPLD.   LOAD MEMORIES.
SPPR.   PRINT MEMORY CONTENTS.
SPREST. ALL OTHER SP COMMANDS.
```

.MCALL. WTSE\$\$,CLEF\$\$.

```
40 ;
41 ;
42 ;
43 ; TABLE OF SP COMMAND MNEMONICS AND THEIR ASSOCIATED
44 ; ROUTINE ADDRESSES.
45 ;
46 ; SPTBL:
47 000000 114 104 .ASCII /LD/ ;LOAD MEMORY.
48 000002 000000G .WORD LD5
49 000004 120 122 .ASCII /PR/ ;PRINT FROM MEMORY.
50 000006 000000G .WORD SPPR5
51 000010 120 123 .ASCII /PS/ ;PAGE SWITCH.
52 000012 000000G .WORD PSS
53 000014 103 114 .ASCII /CL/ ;CALL OR LOADER.
54 000016 000000G .WORD CL5
55 000004 SPNUM = <.-SPTBL>/4
56 ;
57 ;
58 ; CODE FOR MEMORY SELECTION.
59 ;
60 000020 000000 CODE: .WORD 0
```

```

62.      ;
63.      ;
64.      ;      SP ROUTINES.
65.      ;
66.      ;
67. 000022.      SPS::
68. 000022.      CALL.  FIND      ;LOCATE THE COMMAND IN THE COMMAND LINE.
69. 000026. 103003      BCC.  1$      ;OK, CONTINUE.
70. 000030.      CALL.  ERR3
71. 000034. 000422.      BR.  SPSXX.
72. 000036. 022700. 000002      1$:  CMP.  #2,R0      ;COMMANDS ARE 2 CHARS.
73. 000042. 001403      BEQ.  2$
74. 000044.      CALL.  ERR8
75. 000050. 000414      BR.  SPSXX.      ;TRY AGAIN
76.      ;
77. 000052. 012700. 000004      2$:  MOV.  #SPNUM,R0      ;R0 = NUMBER OF COMMANDS.
78. 000056. 012702. 000000*      MOV.  #SPTBL,R2.      ;R2 -> TABLE OF COMMAND MNEMONICS.
79. 000062.      CALL.  SCAN      ;FIND MATCH IN TABLE
80. 000066. 103003      BCC.  3$      ;OK, CONTINUE.
81. 000070.      CALL.  ERR12.      ;COMMAND NOT IN TABLE.
82. 000074. 000402.      BR.  SPSXX.      ;TRY AGAIN
83.      ;
84.      ;      JUMP TO THE ROUTINE THAT GOVERNS THE COMMAND.
85.      ;
86. 000076. 000171. 000000      3$:  JMP.  @R1
87.      ;
88.      ;
89.      ;      LOCAL SP LOOP. INCLUDE PROMPT FOR SP COMMAND.
90.      ;
91.      ;
92. 000102.      SPSXX::
93. 000102. 012767. 050123. 000000C.  MOV.  #*SP.GCMBLK+G.DPRM+2. ;MOVE SP NAME TO GCM BLOCK.
94. 000110.      CALL.  GCONLY.      ;PROMPT.
95. 000114.      CALL.  FIND      ;LOCATE THE COMMAND IN THE COMMAND LINE.
96. 000120. 103003      BCC.  1$      ;OK, CONTINUE.
97. 000122.      CALL.  ERR3
98. 000126. 000765      BR.  SPSXX.
99. 000130. 022700. 000002      1$:  CMP.  #2,R0      ;COMMANDS ARE 2 CHARS.
100. 000134. 001403      BEQ.  2$
101. 000136.      CALL.  ERR8
102. 000142. 000757      BR.  SPSXX.      ;TRY AGAIN
103.      ;
104. 000144. 012700. 000004      2$:  MOV.  #SPNUM,R0      ;R0 = NUMBER OF COMMANDS.
105. 000150. 012702. 000000*      MOV.  #SPTBL,R2.      ;R2 -> TABLE OF COMMAND MNEMONICS.
106. 000154.      CALL.  SCAN      ;FIND MATCH IN TABLE
107. 000160. 103002.      BCC.  3$      ;OK, CONTINUE.
108. 000162. 000167. 000000G.      JMP.  COMXX.      ;RETURN TO 'MAIN', LOOK FOR PROCESSOR MNEMONIC
109.      ;
110.      ;      JUMP TO THE ROUTINE THAT GOVERNS THE COMMAND.
111.      ;
112. 000166. 000171. 000000      3$:  JMP.  @R1

```

SP.....M1110 27-MAR-80 15:31 PAGE 8

Approved For Release 2005/07/11 : CIA-RDP85-00514R000200020001-3

114
115

000001

.END.

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	PLB = 000010	Q\$QLR = 000001
ALUOE = 004000	BYTE43 = 000053	BYTE95 = 000137	PLC = 000020	Q\$QW = 000042
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	PLD = 000030	Q\$RDCD = 000005
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	PLRW = 000200	Q\$RDMD = 000006
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	PLREN = 000200	Q\$REBK = 001000
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	PS5 = 000000	Q\$RNC = 006000
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	Q\$CR1 = 176420	Q\$RSC = 004000
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	Q\$CR2 = 176422	Q\$RSET = 000010
BIT12 = 010000	BYTE50 = 000062	CBKCLK = 000400	Q\$LBR = 176424	Q\$SM = 100000
BIT13 = 020000	BYTE51 = 000063	CLS = 000000	Q\$ATTN = 000100	Q\$SP = 000120
BIT14 = 040000	BYTE52 = 000064	CNOBRE = 100000	Q\$BCL = 000001	Q\$SP2 = 000340
BIT15 = 100000	BYTE53 = 000065	CODE = 000020RG	Q\$CCCP = 000040	RGQEN = 000200
BIT2 = 000004	BYTE54 = 000066	COMXX = 000000	Q\$CHB = 000400	RGQVA = 020000
BIT3 = 000010	BYTE55 = 000067	CPCCEN = 010000	Q\$CHRL = 000200	SCAN = 000000
BIT4 = 000020	BYTE56 = 000070	CPREAD = 040000	Q\$CLR = 000040	SEQCI = 000010
BIT5 = 000040	BYTE57 = 000071	CPWTE = 020000	Q\$CNC = 030000	SPNUM = 000004
BIT6 = 000100	BYTE58 = 000072	CSADRD = 000004	Q\$CP = 000060	SPPRS = 000000
BIT7 = 000200	BYTE59 = 000073	CSEQCI = 100000	Q\$CPCC = 000010	SPS = 000022RG
BIT8 = 000400	BYTE60 = 000074	CSOE = 000040	Q\$CP2 = 000260	SPSXX = 000022RG
BIT9 = 001000	BYTE61 = 000075	CSWTE = 000100	Q\$CSC = 010000	SPTBL = 000000
BYTE0 = 000000	BYTE62 = 000076	DBR RD = 000001	Q\$CSEL = 000360	S\$CLR = 000000
BYTE1 = 000001	BYTE63 = 000077	DB\$CPP = 001457	Q\$CSET = 000002	S\$LA = 000001
BYTE10 = 000012	BYTE64 = 000100	DB\$PTC = 000026	Q\$CSP = 020000	S\$OB = 000005
BYTE11 = 000013	BYTE65 = 000101	DB\$TPC = 000023	Q\$DMA = 000001	S\$OR = 000006
BYTE12 = 000014	BYTE66 = 000102	DISPGS = 100000	Q\$ENBK = 040000	S\$QX = 000004
BYTE13 = 000015	BYTE67 = 000103	DMAAUR = 000005	Q\$ENOP = 020000	S\$SR = 000007
BYTE14 = 000016	BYTE68 = 000104	DMARRD = 000003	Q\$FAL = 004000	S\$STY = 000010
BYTE15 = 000017	BYTE69 = 000105	DMARWR = 000004	Q\$FC = 000045	S\$T = 000014
BYTE16 = 000020	BYTE70 = 000106	ENBR = 010000	Q\$FO = 000044	TD\$CTR = 176370
BYTE17 = 000021	BYTE71 = 000107	ERR12 = 000000	Q\$FP = 000046	TD\$CTW = 176360
BYTE18 = 000022	BYTE72 = 000110	ERR3 = 000000	Q\$HBF = 000002	TD\$INL = 004000
BYTE19 = 000023	BYTE73 = 000111	ERR8 = 000000	Q\$ICP = 000006	TD\$MEM = 000270
BYTE20 = 000024	BYTE74 = 000112	FIND = 000000	Q\$IHB = 000003	TD\$OAR = 176344
BYTE21 = 000025	BYTE75 = 000113	GCMBLK = 000000	Q\$IHRL = 000002	TD\$OTR = 176346
BYTE22 = 000026	BYTE76 = 000114	GCONLY = 000000	Q\$IMRP = 000007	TD\$QRD = 000274
BYTE23 = 000027	BYTE77 = 000115	G.DPRM = 000000	Q\$LBD = 001000	TD\$SW = 176376
BYTE24 = 000030	BYTE78 = 000116	LD5 = 000000	Q\$LBDP = 001001	TD\$STAR = 176372
BYTE25 = 000031	BYTE79 = 000117	LOCEN = 000100	Q\$LBP = 000001	TD\$TAW = 176362
BYTE26 = 000032	BYTE80 = 000120	LOCWA = 040000	Q\$LDCD = 000003	TD\$TDR = 176374
BYTE27 = 000033	BYTE81 = 000121	LOCWB = 100000	Q\$LDMD = 000004	TD\$TDW = 176364
BYTE28 = 000034	BYTE82 = 000122	MAREN1 = 000001	Q\$LDPP = 002000	T\$AD = 000020
BYTE29 = 000035	BYTE83 = 000123	MAREN2 = 004000	Q\$LHP = 010000	T\$BA = 000002
BYTE30 = 000036	BYTE84 = 000124	MARLOD = 010000	Q\$MNC = 140000	T\$BD = 000010
BYTE31 = 000037	BYTE85 = 000125	MAROUT = 000002	Q\$MR = 000052	T\$BSO = 100000
BYTE32 = 000040	BYTE86 = 000126	MARLO = 002000	Q\$MRP = 000040	T\$BT = 000020
BYTE33 = 000041	BYTE87 = 000127	MAROU = 000040	Q\$MRP2 = 000240	T\$BTAR = 000030
BYTE34 = 000042	BYTE88 = 000130	MBKALL = 001000	Q\$MSC = 040000	T\$BTD = 002000
BYTE35 = 000043	BYTE89 = 000131	MBKCLK = 000400	Q\$MSET = 000004	T\$CD = 000100
BYTE36 = 000044	BYTE90 = 000132	MMADRD = 000100	Q\$MSP = 100000	T\$CLK = 002000
BYTE37 = 000045	BYTE91 = 000133	MMLEFT = 000002	Q\$NCLK = 176000	T\$DISK = 000200
BYTE38 = 000046	BYTE92 = 000134	MMOE = 000004	Q\$PP = 000100	T\$DRD = 000004
BYTE39 = 000047	BYTE93 = 000135	MMURTE = 000010	Q\$PPSW = 000320	T\$MEM = 010000
BYTE40 = 000050		MNOBRE = 100000	Q\$PP2 = 000300	T\$FSAR = 000000
BYTE41 = 000051		MREN1 = 000001	Q\$QW = 000043	T\$FSAR = 000004
		MREN2 = 000000	Q\$QLA = 000053	T\$FSAC = 000014
		MSYN = 000040	Q\$QLB = 000054	T\$FSB2 = 000010
		N = 000144		T\$IB = 000026

SP.....M1110 27-MAR-80 15:31 PAGE 8-2
SYMBOL TABLE

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

T\$IBAR= 000024	WORD10= 000024	WORD34= 000104	WORD57= 000162	WORD8 = 000020
T\$IBE= 020000	WORD11= 000026	WORD35= 000106	WORD58= 000164	WORD80= 000240
T\$IBF= 040000	WORD12= 000030	WORD36= 000110	WORD59= 000166	WORD81= 000242
T\$ICD= 000040	WORD13= 000032	WORD37= 000112	WORD6 = 000014	WORD82= 000244
T\$MODE= 004000	WORD14= 000034	WORD38= 000114	WORD60= 000170	WORD83= 000246
T\$OB= 000036	WORD15= 000036	WORD39= 000116	WORD61= 000172	WORD84= 000250
T\$OBE= 004000	WORD16= 000040	WORD4 = 000010	WORD62= 000174	WORD85= 000252
T\$OBF= 010000	WORD17= 000042	WORD40= 000120	WORD63= 000176	WORD86= 000254
T\$OBRA= 000034	WORD18= 000044	WORD41= 000122	WORD64= 000200	WORD87= 000256
T\$OBWA= 000032	WORD19= 000046	WORD42= 000124	WORD65= 000202	WORD88= 000260
T\$OUTA= 100000	WORD2 = 000004	WORD43= 000126	WORD66= 000204	WORD89= 000262
T\$RBD0= 000200	WORD20= 000050	WORD44= 000130	WORD67= 000206	WORD9 = 000022
T\$RNB= 000040	WORD21= 000052	WORD45= 000132	WORD68= 000210	WORD90= 000264
T\$RSET= 040000	WORD22= 000054	WORD46= 000134	WORD69= 000212	WORD91= 000266
T\$SC= 000022	WORD23= 000056	WORD47= 000136	WORD7 = 000016	WORD92= 000270
T\$SCLK= 020000	WORD24= 000060	WORD48= 000140	WORD70= 000214	WORD93= 000272
T\$SEG1= 000000	WORD25= 000062	WORD49= 000142	WORD71= 000216	WORD94= 000274
T\$SEG2= 000001	WORD26= 000064	WORD5 = 000012	WORD72= 000220	WORD95= 000276
T\$SEG3= 000002	WORD27= 000066	WORD50= 000144	WORD73= 000222	WORD96= 000300
T\$SO= 000001	WORD28= 000070	WORD51= 000146	WORD74= 000224	WORD97= 000302
T\$UBUS= 100000	WORD29= 000072	WORD52= 000150	WORD75= 000226	WORD98= 000304
T\$1CLK= 000400	WORD3 = 000006	WORD53= 000152	WORD76= 000230	WORD99= 000306
T\$8BEN= 000020	WORD30= 000074	WORD54= 000154	WORD77= 000232	WORDVAL= 000310
UBD.IN= 000020	WORD31= 000076	WORD55= 000156	WORD78= 000234	XTREAD= 001000
WORD0 = 000000	WORD32= 000100	WORD56= 000160	WORD79= 000236	XTWTE= 000400
WORD1 = 000002	WORD33= 000102			

. ABS. 000000 000
000000 001
SP. 000172 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3210 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:41
SP, SP'-SP=C 20, 1 JIM, C 20, 1 JSP

```

1
2 000000 .TITLE - SPLD
3 .PSECT: SPLD
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23 000000
24 000000 121 130
25 000002 000302
26 000004 123 122
27 000006 000332
28 000010 123 106
29 000012 000322
30 000014 123 060
31 000016 000312
32 000020 123 061
33 000022 000516
34 000024 123 062
35 000026 000526
36 000006
37
38
39
40
41
42
43
44
45
46
47 000030
48 000030
49 000034 103004
50 000036
51 000042 000167 000632
52
53
54
55
56 000046 012700 000006
57 000052 012702 000000

```


Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

58 000056          CALL    SCAN          :MATCH AGAINST COMMAND LINE.
59 000062 103004    BCC     2$           :MATCH WAS MADE.
60 000064          CALL    ERR7          :INVALID MEMORY MNEMONIC.
61 000070 000167 000604 JMP     LD5X
62          :
63          :
64          :
65          :
66          :
67          :
68          :
69          :
70          :
71          :
72          :
73          :
74          :
75          :
76          :
77          :
78          :
79          :
80          :
81          :
82          :
83          :
84 000074 010167 000000G 2$: MOV     R1,RTNPT.      :SAVE POINTER.
85 000100          CALL    FIND          :LOCATE START ADDRESS IN COMMAND LINE.
86 000104 103004    BCC     3$           :OK, CONTINUE.
87 000106          CALL    ERR4          :MISSING OPERAND.
88 000112 000167 000562. JMP     LD5X      :EXIT.
89 000116          CALL    PACK          :CONVERT COMMAND LINE VALUE TO BINARY.
90 000122 103004    BCC     4$           :CONVERSION SUCCESSFUL.
91 000124          CALL    ERR5          :INVALID NUMERIC VALUE.
92 000130 000167 000544 JMP     LD5X
93          :
94 000134 016767 000000G 000000G 4$: MOV     BINWD,MSTRT.  :SAVE LOADING START ADDRESS.
95 000142 016767 000000G 000000G. MOV     BINWD,MSTR2.  :SAVE IT TWICE (FOR REFRESH ON LOOP)
96 000150 012767 177777 000000G. MOV     *-1,MEND.    :INIT END ADDRESS.
97          :
98          :
99          :
100          :
101          :
102 000156          :
103 000162 103004    :
104 000164 052767 000000G 000000G. :
105 000172 000437    :
106          :
107 000174 122711 000114 5$: CMPB     *-L,(R1)      :LOOP INDICATOR.
108 000200 001006    BNE     6$           :NO, MUST BE USPER ADDRESS.
109 000202 016767 000000G 000000G. MOV     MSTR,MEND.    :SET END ADDR = START ADDR.
110 000210          CALL    HANG          :HOW TO STOP LOOP.
111 000214 000426    BR      9$           :JUMP TO RTN.
112          :
113 000216          :
114 000222 103004    6$: CALL     PACK          :CONVERT UPPER ADDRESS.
                   BCC     7$           :OK, CONTINUE.

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
115 000224
116 000230 000167 000444 CALL ERRS : INVALID NUMERIC
117 JMP LDSX : EXIT
118
119 : SAVE END ADDRESS (BINARY)
120 : CHECK FOR LOOP INDICATOR AFTER END ADDRESS (CONDITION 4)
121 000234 016767 000000G 000000G 7$: MOV BINWD,MEND : SET UP END ADDRESS
122 000242 CALL FIND : CHECK FOR LOOP INDICATOR
123 000246 103411 BCS 9$ : NO LOOP
124 000250 122711 000114 CMPB #'L,(R1) : CORRECT INDICATOR
125 000254 001404 BEQ 8$ : YES, CONTINUE
126 000256 CALL ERR11 : LOOP OPTION ERROR
127 000262 000167 000412 JMP LDSX
128 000266 8$: CALL HANG : HOW TO STOP LOOP
129
130 000272 016701 000000G 9$: MOV RTNPT,R1 : POINT TO ROUTINE
131 000276 000171 000000 JMP @R1 : JUMP TO ROUTINE
132
133 :
134 :
135 : LOAD QEX MEMORY
136 000302 012767 000004 000000G LD5QX: MOV #S$QX, CODE : SET MEMORY SELECT CODE = QEX
137 000310 000420 BR LDSQ
138
139 :
140 : LOAD SDLB PAGE MEMORY
141 000312 012767 000005 000000G LD5QB: MOV #S$QB, CODE : SET MEMORY SELECT CODE = SDLB PAGE
142 000320 000414 BR LDSQ
143
144 :
145 : LOAD QLB REFERENCE PAGE
146 000322 LD5QR:
147 000322 012767 000006 000000G MOV #S$QR, CODE : SET MEMORY SELECT CODE = REFERENCE PAGE
148 000330 000410 BR LDSQ
149
150 :
151 : LOAD SUBREAD MEMORY
152 000332 LD5SR:
153 000332 012767 000007 000000G MOV #S$SR, CODE
154 000340 016746 000000G MOV SRHIGH, -(SP) : SUPPLY UPPER MEMORY LIMIT
155 000344 016746 000000G MOV SRLow, -(SP) : LOWER LIMIT
156 000350 000404 BR LDSG
157 000352 LD5Q:
158 000352 016746 000000G MOV SQHIGH, -(SP) : SUPPLY UPPER MEMORY LIMIT
159 000356 016746 000000G MOV SQLOW, -(SP) : LOWER LIMIT
160 000362 LD5G:
161 000362 CALL BUFSET : PREPARE FOR LOAD
162 000366 103002 BCC 1$ : OK, CONTINUE
163 000370 000167 000304 JMP LDSX : ERROR, EXIT
164
165 000374 032767 000000G 000000G 1$: BIT #RP, BASE : REPEAT PROMPT
166 000402 001414 BEQ 5$ : NO, ONCE ONLY
167 000404 2$: CALL PDATA : READ DATA FROM COMMAND LINE
168 000410 103401 BCS 3$ : END OF MEMORY
169 000412 102002 BVC 4$ : NO <CR> RESPONSE, CONTINUE
170 000414 3$:
171 000414 000167 000260 JMP LDSX
```

```
172. ;
173 000420 4$:
174 000420
175 000424 066767 000000G.000000G. CALL LODSM1 ;LOAD 16 BIT MEMORY.
176 000432 000764 ADD INCVAL,MSTR2. ;BUMP ADDRESS.
177 ; BR 2$ ;REPEAT.
178 ;
179 ; PROMPT ONCE THEN FILL MEMORY.
180 000434 5$: CALL PDATA. ;PROMPT.
181 000440 103401 BCS 6$ ;END OF MEMORY.
182 000442 102002 BVC 7$ ;NO <CR> RESPONSE. CONTINUE.
183 000444 6$:
184 000444 000167 000230 JMP LD5X
185 ;
186 000450 7$: CALL LODSM1 ;LOAD 16 BIT MEMORY.
187 000450
188 ;
189 000454 066767 000000G.000000G. ADD INCVAL,MSTR2. ;ADVANCE ADDRESS.
190 000462 026767 000000G.000000G. CMP MSTR2,MEND. ;HAS UPPER MEMORY LIMIT BEEN REACHED.
191 000470 101767 BLOS 7$ ;NO. CONTINUE.
192 000472 032767 000000G.000000G. BIT #LOOP,BASE. ;LOOP ON.
193 000500 001002 BNE 8$ ;YES. CONTINUE.
194 000502 000167 000172 JMP LD5X
195 000506 016767 000000G.000000G.8$ MOV MSTRT,MSTR2. ;REINITIALIZE ADDRESS.
196 000514 000755 BR 7$
```

```

198 ; LOAD SIDMEM (MULTIPLE PAGES)
199 ;
200 LD5S1::
201 000516 012767 000010 000000G MOV #S$S1.CODE ;SELECT SIDMEM 1
202 000524 000403 BR LD5S
203 000526 LD5S2::
204 000526 012767 000014 000000G MOV #S$S2.CODE ;SELECT SIDMEM 2
205 000534 LD5S:
206 000534 016746 000000G MOV SDHIGH, -(SP) ;SUPPLY UPPER MEMORY LIMIT
207 000540 016746 000000G MOV SDLOW, -(SP) ;LOWER LIMIT
208 000544 CALL BUFS3 ;PREPARE FOR LOAD
209 000550 103002 BCC 1$ ;OK, CONTINUE
210 000552 000167 000122 JMP LD5X ;ERROR, EXIT
211 ;
212 000556 032767 000000G 1$: BIT #RP.BASE ;REPEAT PROMPT
213 000564 001414 BEQ 5$ ;NO, ONCE ONLY
214 000566 2$: CALL PDATA ;READ DATA FROM COMMAND LINE
215 000572 103401 BCS 3$ ;END OF MEMORY
216 000574 102002 BVC 4$ ;NO <CR> RESPONSE, CONTINUE
217 000576 3$:
218 000576 000167 000076 JMP LD5X
219 ;
220 000602 4$:
221 000602 CALL LOD5M3 ;LOAD 48 BIT MEMORY
222 ;
223 000606 066767 000000G 000000G ADD INCVAL, MSTR2 ;BUMP ADDRESS
224 000614 000764 BR 2$ ;REPEAT
225 ;
226 ; PROMPT ONCE THEN FILL MEMORY
227 ;
228 000616 5$: CALL PDATA ;PROMPT
229 000622 103401 BCS 6$ ;END OF MEMORY
230 000624 102002 BVC 7$ ;NO <CR> RESPONSE, CONTINUE
231 000626 6$:
232 000626 000167 000046 JMP LD5X
233 ;
234 000632 7$:
235 000632 CALL LOD5M3 ;LOAD 48 BIT MEMORY
236 ;
237 000636 066767 000000G 000000G ADD INCVAL, MSTR2 ;ADVANCE ADDRESS
238 000644 026767 000000G 000000G CMP MSTR2, MEND ;HAS UPPER MEMORY LIMIT BEEN REACHED
239 000652 101767 BLOS 7$ ;NO, CONTINUE
240 000654 032767 000000G 000000G BIT #LOOP.BASE ;LOOP ON
241 000662 001002 BNE 8$ ;YES, CONTINUE
242 000664 000167 000010 JMP LD5X
243 000670 016767 000000G 000000G 8$: MOV MSTR, MSTR2 ;REINITIALIZE ADDRESS
244 000676 000755 BR 7$
245 ;
246 ;
247 000700 LD5X:
248 000700 012746 000000 MOV #S$CLR, -(SP) ;CLEAR SP
249 000704 CALL SPCR ;WRITE TO CONTROL REG
250 000710 042767 000000G 000000G BIC #RP.BASE ;CLEAR PROMPT FLAG
251 000716 CALL KILL ;KILL AST (IF THERE WAS ONE)
252 000722 000167 000000G JMP SPSXX

```

```
254      ;
255      ;      LOAD SINGLE WORD MEMORY.
256      ;
257 000726      LODSM1:
258 000726 012746 000001      MOV      #S$LA, -(SP)      ; ADDRESS SELECT.
259 000732      CALL      SPCR      ; SEND TO SP CONTROL REG.
260 000736 016746 000000G      MOV      MSTR2, -(SP)      ; ACTUAL ADDRESS.
261 000742      CALL      LBSP      ; SEND TO SP.
262 000746 016746 000000G      MOV      CODE, -(SP)      ; SELECT MEMORY.
263 000752      CALL      SPCR
264 000756 016746 000000G      MOV      DATA1, -(SP)      ; DATA WORD FOR MEMORY.
265 000762      CALL      LBSP      ; SEND DATA TO SP.
266 000766      RETURN.
267      ;
268      ;      LOAD 3 WORD MEMORY.
269      ;
270 000770      LODSM3:
271 000770      SAVE      CODE, DATA1
272 001000      CALL      LODSM1      ; LOAD WORD A.
273 001004 005267 000000G      INC      CODE      ; SELECT WORD B.
274 001010 016767 000000G 000000G      MOV      DATA2, DATA1      ; SET UP DATA1
275 001016      CALL      LODSM1      ; LOAD WORD B.
276 001022 005267 000000G      INC      CODE      ; SELECT WORD C.
277 001026 016767 000000G 000000G      MOV      DATA3, DATA1      ; SET UP DATA1
278 001034      CALL      LODSM1      ; LOAD WORD C.
279 001040      RESTOR      CODE, DATA1
280 001050      RETURN.
281      ;
282      000001      .END
```

```
ALUCKE = 040000
ALUOE = 004000
A01 = 010000
BASE = ***** GX
BINWD = ***** GX
BITVAL = 000000
BIT0 = 000001
BIT1 = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2 = 000004
BIT3 = 000010
BIT4 = 000020
BIT5 = 000040
BIT6 = 000100
BIT7 = 000200
BIT8 = 000400
BIT9 = 001000
BUFSET = ***** GX
BUFS3 = ***** GX
BYTE0 = 000000
BYTE1 = 000001
BYTE10 = 000012
BYTE11 = 000013
BYTE12 = 000014
BYTE13 = 000015
BYTE14 = 000016
BYTE15 = 000017
BYTE16 = 000020
BYTE17 = 000021
BYTE18 = 000022
BYTE19 = 000023
BYTE2 = 000002
BYTE20 = 000024
BYTE21 = 000025
BYTE22 = 000026
BYTE23 = 000027
BYTE24 = 000030
BYTE25 = 000031
BYTE26 = 000032
BYTE27 = 000033
BYTE28 = 000034
BYTE29 = 000035
BYTE3 = 000003
BYTE30 = 000036
BYTE31 = 000037
BYTE32 = 000040
BYTE33 = 000041
BYTE34 = 000042
BYTE35 = 000043
BYTE36 = 000044
BYTE37 = 000045
BYTE38 = 000046
BYTE39 = 000047
BYTE4 = 000004
BYTE40 = 000050
BYTE41 = 000051
BYTE42 = 000052
BYTE43 = 000053
BYTE44 = 000054
BYTE45 = 000055
BYTE46 = 000056
BYTE47 = 000057
BYTE48 = 000060
BYTE49 = 000061
BYTE5 = 000005
BYTE50 = 000062
BYTE51 = 000063
BYTE52 = 000064
BYTE53 = 000065
BYTE54 = 000066
BYTE55 = 000067
BYTE56 = 000070
BYTE57 = 000071
BYTE58 = 000072
BYTE59 = 000073
BYTE6 = 000006
BYTE60 = 000074
BYTE61 = 000075
BYTE62 = 000076
BYTE63 = 000077
BYTE64 = 000100
BYTE65 = 000101
BYTE66 = 000102
BYTE67 = 000103
BYTE68 = 000104
BYTE69 = 000105
BYTE7 = 000007
BYTE70 = 000106
BYTE71 = 000107
BYTE72 = 000110
BYTE73 = 000111
BYTE74 = 000112
BYTE75 = 000113
BYTE76 = 000114
BYTE77 = 000115
BYTE78 = 000116
BYTE79 = 000117
BYTE8 = 000010
BYTE80 = 000120
BYTE81 = 000121
BYTE82 = 000122
BYTE83 = 000123
BYTE84 = 000124
BYTE85 = 000125
BYTE86 = 000126
BYTE87 = 000127
BYTE88 = 000130
BYTE89 = 000131
BYTE9 = 000011
BYTE90 = 000132
BYTE91 = 000133
BYTE92 = 000134
BYTE93 = 000135
BYTE94 = 000136
BYTE95 = 000137
BYTE96 = 000140
BYTE97 = 000141
BYTE98 = 000142
BYTE99 = 000143
BYTVAL = 000144
CBKALL = 001000
CBKCLK = 000400
CNOBRE = 100000
CODE = ***** GX
CPCCEN = 010000
CPREAD = 040000
CPWRTE = 020000
CSADRD = 000004
CSEQCI = 100000
CSOE = 000040
CSWRTE = 000100
DATA1 = ***** GX
DATA2 = ***** GX
DATA3 = ***** GX
DBR.RD = 000001
DB*CPP = 001457
DB*SPT = 000026
DB*TPC = 000023
DISPGS = 100000
DMAWR = 000005
DMARRD = 000003
DMARWR = 000004
ENBR = 010000
ERR11 = ***** GX
ERR4 = ***** GX
ERP5 = ***** GX
ERR7 = ***** GX
FIND = ***** GX
HANG = ***** GX
INCVAL = ***** GX
KILL = ***** GX
LBSP = ***** GX
LD5 = 000030RG
LD5G0 = 000362R
LD5LN = 000006 G
LD5Q = 000352R
LD5QB = 000312RG
LD5QB = 000322RG
LD5QX = 000302RG
LD5S = 000534R
LD5SR = 000332RG
LD5S1 = 000516RG
LD5S2 = 000526RG
LD5TBL = 000000RG
LD5X = 000700R
LOC.EN = 000100
LOC.WA = 040000
LOC.WB = 100000
LDSM1 = 000726R
LDSM3 = 000770R
LOOP = ***** GX
MAREN1 = 000001
MAREN2 = 004000
MARLOD = 010000
MAROUT = 000002
MAR.LO = 002000
MAR.BU = 000040
MBKALL = 001000
MBKCLK = 000400
MEND = ***** GX
MHARD = 000100
MMLEFT = 000002
MMOE = 000004
MMWRTE = 000010
MNOBRE = 100000
MREN1 = 000001
MREN2 = 020000
MSTRT = ***** GX
MSTR2 = ***** GX
MSYN = 000040
N = 000144
PACK = ***** GX
PDATA = ***** GX
PLB = 000010
PLC = 000020
PLD = 000030
PLRWR = 000200
PLR.EN = 000200
QR*CR1 = 176420
QR*CR2 = 176422
QR*LBR = 176424
Q*ATTN = 000100
Q*BC = 000001
Q*CCCP = 000040
Q*CHB = 000400
Q*CHRL = 000200
Q*CLR = 000040
Q*CN = 030000
Q*CP = 000060
Q*CPCC = 000000
Q*CP2 = 000260
Q*CS = 010000
Q*CSEL = 000360
Q*CSET = 000002
Q*QCS = 020000
Q*DMA = 000001
Q*ENBK = 040000
Q*ENOP = 020000
Q*FAL = 004000
Q*F = 000045
Q*FO = 000044
Q*FP = 000046
Q*H = 000043
Q*ICP = 000006
Q*IHB = 000003
Q*IHLR = 000002
Q*IMRP = 000007
Q*LBD = 001000
Q*LBPD = 001001
Q*LBP = 000001
Q*LCD = 000003
Q*LDMD = 000004
Q*LDPP = 002000
Q*LHP = 010000
Q*LMC = 140000
Q*MR = 000052
Q*MRP = 000040
Q*MRP2 = 000240
Q*MSC = 040000
Q*MSET = 000004
Q*MSP = 100000
Q*NCCLK = 176000
Q*PP = 000100
Q*PPSW = 000320
Q*PP2 = 000300
Q*OHLT = 000013
Q*QL = 000043
Q*QLA = 000053
Q*QLB = 000054
Q*QLR = 000001
Q*QW = 000042
Q*RD = 000005
Q*RDMD = 000006
Q*REBK = 001000
Q*RN = 006000
Q*RS = 004000
Q*RSET = 000010
Q*SM = 100000
Q*SP = 000120
Q*SP2 = 000340
RGQ.EN = 000200
RGQ.VA = 020000
RP = ***** GX
RTNPT = ***** GX
SCAN = ***** GX
SDHIGH = ***** GX
SDLOW = ***** GX
SEQ.CF = 000010
SPCR = ***** GX
SPSXX = ***** GX
SQHIGH = ***** GX
SQLOW = ***** GX
SRHIGH = ***** GX
SRLOW = ***** GX
S*CLR = 000000
S*LA = 000001
S*QB = 000005
S*QR = 000006
S*QX = 000004
S*SR = 000007
```

SPLD---MACRO-M1110 27-MAR-80 15:32 PAGE 7-2
SYMBOL TABLE

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

S\$S1 = 000010	T\$IBE = 020000	WORD17 = 000042	WORD46 = 000134	WORD74 = 000224
S\$S2 = 000014	T\$IBF = 040000	WORD18 = 000044	WORD47 = 000136	WORD75 = 000226
TD\$CTR = 176370	T\$ICD = 000040	WORD19 = 000046	WORD48 = 000140	WORD76 = 000230
TD\$CTW = 176360	T\$MODE = 004000	WORD2 = 000004	WORD49 = 000142	WORD77 = 000232
TD\$INL = 004000	T\$OB = 000036	WORD20 = 000050	WORDS = 000012	WORD78 = 000234
TD\$MEM = 000270	T\$OBE = 004000	WORD21 = 000052	WORD50 = 000144	WORD79 = 000236
TD\$OAR = 176344	T\$OBF = 010000	WORD22 = 000054	WORD51 = 000146	WORD8 = 000020
TD\$OTR = 176346	T\$OBRA = 000034	WORD23 = 000056	WORD52 = 000150	WORD80 = 000240
TD\$ORD = 000274	T\$OBWA = 000032	WORD24 = 000060	WORD53 = 000152	WORD81 = 000242
TD\$SW = 176376	T\$OUTA = 100000	WORD25 = 000062	WORD54 = 000154	WORD82 = 000244
TD\$TAR = 176372	T\$RBD0 = 000200	WORD26 = 000064	WORD55 = 000156	WORD83 = 000246
TD\$TAW = 176362	T\$RNB = 000040	WORD27 = 000066	WORD56 = 000160	WORD84 = 000250
TD\$TDR = 176374	T\$RSET = 040000	WORD28 = 000070	WORD57 = 000162	WORD85 = 000252
TD\$TDW = 176364	T\$SC = 000022	WORD29 = 000072	WORD58 = 000164	WORD86 = 000254
TD\$AD = 000020	T\$SCLK = 020000	WORD3 = 000006	WORD59 = 000166	WORD87 = 000256
T\$BA = 000002	T\$SEG1 = 000000	WORD30 = 000074	WORD6 = 000014	WORD88 = 000260
T\$BD = 000010	T\$SEG2 = 000001	WORD31 = 000076	WORD60 = 000170	WORD89 = 000262
T\$BS0 = 100000	T\$SEG3 = 000002	WORD32 = 000100	WORD61 = 000172	WORD9 = 000022
T\$BT = 000020	T\$S0 = 000001	WORD33 = 000102	WORD62 = 000174	WORD90 = 000264
T\$BTAR = 000030	T\$UBUS = 100000	WORD34 = 000104	WORD63 = 000176	WORD91 = 000266
T\$BDT = 000200	T\$1CLK = 000400	WORD35 = 000106	WORD64 = 000200	WORD92 = 000270
T\$CD = 000100	T\$BEN = 000020	WORD36 = 000110	WORD65 = 000202	WORD93 = 000272
T\$CLK = 000200	UBD IN = 000020	WORD37 = 000112	WORD66 = 000204	WORD94 = 000274
T\$DISK = 000200	WORD0 = 000000	WORD38 = 000114	WORD67 = 000206	WORD95 = 000276
T\$DRD = 000004	WORD1 = 000002	WORD39 = 000116	WORD68 = 000210	WORD96 = 000300
T\$EMEM = 010000	WORD10 = 000024	WORD4 = 000010	WORD69 = 000212	WORD97 = 000302
T\$FSAA = 000000	WORD11 = 000026	WORD40 = 000120	WORD7 = 000016	WORD98 = 000304
T\$FSAB = 000004	WORD12 = 000030	WORD41 = 000122	WORD70 = 000214	WORD99 = 000306
T\$FSAC = 000014	WORD13 = 000032	WORD42 = 000124	WORD71 = 000216	WRDVAL = 000310
T\$FSB2 = 000010	WORD14 = 000034	WORD43 = 000126	WORD72 = 000220	XTREAD = 001000
T\$IB = 000026	WORD15 = 000036	WORD44 = 000130	WORD73 = 000222	XTWTE = 000400
T\$IBAR = 000024	WORD16 = 000040	WORD45 = 000132		

. ABS. 000000 000
000000 001
SPLD. 001052 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3456 WORDS (14 PAGES)
DYNAMIC MEMORY: 4916 WORDS (18 PAGES)
ELAPSED TIME: 00:00:46
SPLD, SPLD/SP=C20.11IM, C20.13SPLD

```

1      .TITLE- SPPR
2 000000 .PSECT- SPPR
3
4      :
5      :
6      :   HARDWARE-QUERY-RESOLVER- "MANUAL" DEBUGGING AIDS-
7      :   SUBDOCUMENT-PROCESSOR-TEST-ROUTINES-
8      :
9      :   COMMANDS:
10     PR      PRINT-SP-MEMORIES-
11
12     :
13     :   ONCE-A-COMMAND-HAS-BEEN-EXECUTED (OR-AN-ERROR-ENCOUNTERED)
14     :   THIS-MODULE-RETURNS-CONTROL-TO- THE-MODULE-SP-AT-LOCATION-
15     :   "SPXX".
16     :
17     :
18     :   .MCALL- WTSE$S,CLEF$S-
19     :
20     :
21     :   TABLE-OF-VALID-SP-MEMORY-MNEMONICS-AND-ADDRESSES-OF- THEIR-
22     :   ASSOCIATED-ROUTINES-
23
24 000000 PR5TBL::
25 000000 121 130 .ASCII- /OX/ :DEX-MEMORY-
26 000000 123 122 .WORD- PR5OX-
27 000000 000346 106 .ASCII- /SR/ :SUBREAD-MEMORY-
28 000010 123 .WORD- PR5SR-
29 000012 000336 .ASCII- /SF/ :SDLB-REFERENCE-PAGE-
30 000014 123 060 .WORD- PR5OR-
31 000016 000326 .ASCII- /S0/ :SDLB-PAGE-
32 000020 123 061 .WORD- PR5OB-
33 000022 000470 .ASCII- /S1/ :SIDMEM1-
34 000024 123 062 .WORD- PR5S1
35 000026 000500 .ASCII- /S2/ :SIDMEM2-
36 000030 .WORD- PR5S2-
37 PR5LN- == <.-PR5TBL>
38
39 :
40 :
41 :   PRINT-
42 :   PERFORM-THIRD-LEVEL-PARSING-
43 :   EG. IN- THE-COMMAND:
44 :   SP>PR-OX-0
45 :   PARSE- THE- "OX"
46 :
47 000030 SPPR5::
48 000030 CALL- FIND :LOCATE-MEMORY-MNEMONIC-IN-COMMAND-LINE-
49 000034 103004 BCC- 1$ :OK- CONTINUE-
50 000036 CALL- ERR4 :MISSING-OPERAND-
51 000042 000167 000614 JMP- PR5X :EXIT-
52
53 :
54 :   MATCH- THE-MNEMONIC- IN- THE-COMMAND-LINE- AGAINST- THE-TABLE-
55 :   OF- VALID-MNEMONICS-
56
57 000046 012700 000030 1$: MOV- #PR5LN,R0 :NUMBER-OF-TABLE-ENTRIES-
58 000052 012702 000000 MOV- #PR5TBL,R2 :PRINT-TO-TABLE-

```



```

58 000056          CALL    SCAN          ;MATCH AGAINST COMMAND LINE
59 000062 103004    BCC     2$           ;MATCH WAS MADE
60 000064          CALL    ERR7          ;INVALID MEMORY MNEMONIC
61 000070 000167 000566 JMP     PR5X
62          ;
63          ;
64          ;
65          ;
66          ;
67          ;
68          ;
69          ;
70          ;
71          ;
72          ;
73          ;
74          ;
75          ;
76          ;
77          ;
78          ;
79          ;
80          ;
81          ;
82          ;
83          ;
84 000074 010167 000000G 2$: MOV     R1,R1NPT ;SAVE POINTER
85 000100          CALL    FIND          ;LOCATE START ADDRESS IN COMMAND LINE
86 000104 103004    BCC     3$           ;OK, CONTINUE
87 000106          CALL    ERR4          ;MISSING OPERAND
88 000112 000167 000544 JMP     PR5X ;EXIT
89 000116          CALL    PACK          ;CONVERT COMMAND LINE VALUE TO BINARY
90 000122 103004    BCC     4$           ;CONVERSION SUCCESSFUL
91 000124          CALL    ERR5          ;INVALID NUMERIC VALUE
92 000130 000167 000526 JMP     PR5X
93          ;
94 000134 016767 000000G 000000G 4$: MOV     BINWD,MSTR2 ;SAVE LOADING START ADDRESS
95 000142 016767 000000G 000000G MOV     BINWD,MSTR2 ;SAVE IT TWICE (FOR REFRESH ON LOOP)
96 000150 012767 177777 000000G MOV     *-1,MEND ;INIT END ADDRESS
97          ;
98          ;
99          ;
100          ;
101          ;
102 000156          ;
103 000162 103004    CALL    FIND          ;SCAN COMMAND LINE
104 000164 016767 000000G 000000G BCC     5$           ;SOMETHING THERE
105 000172 000445    MOV     MSTR2,MEND ;SET END ADDR = START ADDR
106          BR      9$
107 000174 122711 000114 5$: GMPB     *-L,(R1) ;LOOP INDICATOR
108 000200 001011    BNE     6$           ;NO, MUST BE USPER ADDRESS
109 000202 052767 000000G 000000G BIS     #OUT,BASE ;SET FLAG FOR OUTPUT CONTROL
110 000210 016767 000000G 000000G MOV     MSTR2,MEND ;SET END ADDR = START ADDR
111 000216          CALL    HANG          ;HOW TO STOP LOOP
112 000222 000431    BR      9$           ;JUMP TO RTN
113          ;
114 000224          ;
115          ;
116          ;
117          ;
118          ;
119          ;
120          ;
121          ;
122          ;
123          ;
124          ;
125          ;
126          ;
127          ;
128          ;
129          ;
130          ;
131          ;
132          ;
133          ;
134          ;
135          ;
136          ;
137          ;
138          ;
139          ;
140          ;
141          ;
142          ;
143          ;
144          ;
145          ;
146          ;
147          ;
148          ;
149          ;
150          ;
151          ;
152          ;
153          ;
154          ;
155          ;
156          ;
157          ;
158          ;
159          ;
160          ;
161          ;
162          ;
163          ;
164          ;
165          ;
166          ;
167          ;
168          ;
169          ;
170          ;
171          ;
172          ;
173          ;
174          ;
175          ;
176          ;
177          ;
178          ;
179          ;
180          ;
181          ;
182          ;
183          ;
184          ;
185          ;
186          ;
187          ;
188          ;
189          ;
190          ;
191          ;
192          ;
193          ;
194          ;
195          ;
196          ;
197          ;
198          ;
199          ;
200          ;
201          ;
202          ;
203          ;
204          ;
205          ;
206          ;
207          ;
208          ;
209          ;
210          ;
211          ;
212          ;
213          ;
214          ;
215          ;
216          ;
217          ;
218          ;
219          ;
220          ;
221          ;
222          ;
223          ;
224          ;
225          ;
226          ;
227          ;
228          ;
229          ;
230          ;
231          ;
232          ;
233          ;
234          ;
235          ;
236          ;
237          ;
238          ;
239          ;
240          ;
241          ;
242          ;
243          ;
244          ;
245          ;
246          ;
247          ;
248          ;
249          ;
250          ;
251          ;
252          ;
253          ;
254          ;
255          ;
256          ;
257          ;
258          ;
259          ;
260          ;
261          ;
262          ;
263          ;
264          ;
265          ;
266          ;
267          ;
268          ;
269          ;
270          ;
271          ;
272          ;
273          ;
274          ;
275          ;
276          ;
277          ;
278          ;
279          ;
280          ;
281          ;
282          ;
283          ;
284          ;
285          ;
286          ;
287          ;
288          ;
289          ;
290          ;
291          ;
292          ;
293          ;
294          ;
295          ;
296          ;
297          ;
298          ;
299          ;
300          ;
301          ;
302          ;
303          ;
304          ;
305          ;
306          ;
307          ;
308          ;
309          ;
310          ;
311          ;
312          ;
313          ;
314          ;
315          ;
316          ;
317          ;
318          ;
319          ;
320          ;
321          ;
322          ;
323          ;
324          ;
325          ;
326          ;
327          ;
328          ;
329          ;
330          ;
331          ;
332          ;
333          ;
334          ;
335          ;
336          ;
337          ;
338          ;
339          ;
340          ;
341          ;
342          ;
343          ;
344          ;
345          ;
346          ;
347          ;
348          ;
349          ;
350          ;
351          ;
352          ;
353          ;
354          ;
355          ;
356          ;
357          ;
358          ;
359          ;
360          ;
361          ;
362          ;
363          ;
364          ;
365          ;
366          ;
367          ;
368          ;
369          ;
370          ;
371          ;
372          ;
373          ;
374          ;
375          ;
376          ;
377          ;
378          ;
379          ;
380          ;
381          ;
382          ;
383          ;
384          ;
385          ;
386          ;
387          ;
388          ;
389          ;
390          ;
391          ;
392          ;
393          ;
394          ;
395          ;
396          ;
397          ;
398          ;
399          ;
400          ;
401          ;
402          ;
403          ;
404          ;
405          ;
406          ;
407          ;
408          ;
409          ;
410          ;
411          ;
412          ;
413          ;
414          ;
415          ;
416          ;
417          ;
418          ;
419          ;
420          ;
421          ;
422          ;
423          ;
424          ;
425          ;
426          ;
427          ;
428          ;
429          ;
430          ;
431          ;
432          ;
433          ;
434          ;
435          ;
436          ;
437          ;
438          ;
439          ;
440          ;
441          ;
442          ;
443          ;
444          ;
445          ;
446          ;
447          ;
448          ;
449          ;
450          ;
451          ;
452          ;
453          ;
454          ;
455          ;
456          ;
457          ;
458          ;
459          ;
460          ;
461          ;
462          ;
463          ;
464          ;
465          ;
466          ;
467          ;
468          ;
469          ;
470          ;
471          ;
472          ;
473          ;
474          ;
475          ;
476          ;
477          ;
478          ;
479          ;
480          ;
481          ;
482          ;
483          ;
484          ;
485          ;
486          ;
487          ;
488          ;
489          ;
490          ;
491          ;
492          ;
493          ;
494          ;
495          ;
496          ;
497          ;
498          ;
499          ;
500          ;
501          ;
502          ;
503          ;
504          ;
505          ;
506          ;
507          ;
508          ;
509          ;
510          ;
511          ;
512          ;
513          ;
514          ;
515          ;
516          ;
517          ;
518          ;
519          ;
520          ;
521          ;
522          ;
523          ;
524          ;
525          ;
526          ;
527          ;
528          ;
529          ;
530          ;
531          ;
532          ;
533          ;
534          ;
535          ;
536          ;
537          ;
538          ;
539          ;
540          ;
541          ;
542          ;
543          ;
544          ;
545          ;
546          ;
547          ;
548          ;
549          ;
550          ;
551          ;
552          ;
553          ;
554          ;
555          ;
556          ;
557          ;
558          ;
559          ;
560          ;
561          ;
562          ;
563          ;
564          ;
565          ;
566          ;
567          ;
568          ;
569          ;
570          ;
571          ;
572          ;
573          ;
574          ;
575          ;
576          ;
577          ;
578          ;
579          ;
580          ;
581          ;
582          ;
583          ;
584          ;
585          ;
586          ;
587          ;
588          ;
589          ;
590          ;
591          ;
592          ;
593          ;
594          ;
595          ;
596          ;
597          ;
598          ;
599          ;
600          ;
601          ;
602          ;
603          ;
604          ;
605          ;
606          ;
607          ;
608          ;
609          ;
610          ;
611          ;
612          ;
613          ;
614          ;
615          ;
616          ;
617          ;
618          ;
619          ;
620          ;
621          ;
622          ;
623          ;
624          ;
625          ;
626          ;
627          ;
628          ;
629          ;
630          ;
631          ;
632          ;
633          ;
634          ;
635          ;
636          ;
637          ;
638          ;
639          ;
640          ;
641          ;
642          ;
643          ;
644          ;
645          ;
646          ;
647          ;
648          ;
649          ;
650          ;
651          ;
652          ;
653          ;
654          ;
655          ;
656          ;
657          ;
658          ;
659          ;
660          ;
661          ;
662          ;
663          ;
664          ;
665          ;
666          ;
667          ;
668          ;
669          ;
670          ;
671          ;
672          ;
673          ;
674          ;
675          ;
676          ;
677          ;
678          ;
679          ;
680          ;
681          ;
682          ;
683          ;
684          ;
685          ;
686          ;
687          ;
688          ;
689          ;
690          ;
691          ;
692          ;
693          ;
694          ;
695          ;
696          ;
697          ;
698          ;
699          ;
700          ;
701          ;
702          ;
703          ;
704          ;
705          ;
706          ;
707          ;
708          ;
709          ;
710          ;
711          ;
712          ;
713          ;
714          ;
715          ;
716          ;
717          ;
718          ;
719          ;
720          ;
721          ;
722          ;
723          ;
724          ;
725          ;
726          ;
727          ;
728          ;
729          ;
730          ;
731          ;
732          ;
733          ;
734          ;
735          ;
736          ;
737          ;
738          ;
739          ;
740          ;
741          ;
742          ;
743          ;
744          ;
745          ;
746          ;
747          ;
748          ;
749          ;
750          ;
751          ;
752          ;
753          ;
754          ;
755          ;
756          ;
757          ;
758          ;
759          ;
760          ;
761          ;
762          ;
763          ;
764          ;
765          ;
766          ;
767          ;
768          ;
769          ;
770          ;
771          ;
772          ;
773          ;
774          ;
775          ;
776          ;
777          ;
778          ;
779          ;
780          ;
781          ;
782          ;
783          ;
784          ;
785          ;
786          ;
787          ;
788          ;
789          ;
790          ;
791          ;
792          ;
793          ;
794          ;
795          ;
796          ;
797          ;
798          ;
799          ;
800          ;
801          ;
802          ;
803          ;
804          ;
805          ;
806          ;
807          ;
808          ;
809          ;
810          ;
811          ;
812          ;
813          ;
814          ;
815          ;
816          ;
817          ;
818          ;
819          ;
820          ;
821          ;
822          ;
823          ;
824          ;
825          ;
826          ;
827          ;
828          ;
829          ;
830          ;
831          ;
832          ;
833          ;
834          ;
835          ;
836          ;
837          ;
838          ;
839          ;
840          ;
841          ;
842          ;
843          ;
844          ;
845          ;
846          ;
847          ;
848          ;
849          ;
850          ;
851          ;
852          ;
853          ;
854          ;
855          ;
856          ;
857          ;
858          ;
859          ;
860          ;
861          ;
862          ;
863          ;
864          ;
865          ;
866          ;
867          ;
868          ;
869          ;
870          ;
871          ;
872          ;
873          ;
874          ;
875          ;
876          ;
877          ;
878          ;
879          ;
880          ;
881          ;
882          ;
883          ;
884          ;
885          ;
886          ;
887          ;
888          ;
889          ;
890          ;
891          ;
892          ;
893          ;
894          ;
895          ;
896          ;
897          ;
898          ;
899          ;
900          ;
901          ;
902          ;
903          ;
904          ;
905          ;
906          ;
907          ;
908          ;
909          ;
910          ;
911          ;
912          ;
913          ;
914          ;
915          ;
916          ;
917          ;
918          ;
919          ;
920          ;
921          ;
922          ;
923          ;
924          ;
925          ;
926          ;
927          ;
928          ;
929          ;
930          ;
931          ;
932          ;
933          ;
934          ;
935          ;
936          ;
937          ;
938          ;
939          ;
940          ;
941          ;
942          ;
943          ;
944          ;
945          ;
946          ;
947          ;
948          ;
949          ;
950          ;
951          ;
952          ;
953          ;
954          ;
955          ;
956          ;
957          ;
958          ;
959          ;
960          ;
961          ;
962          ;
963          ;
964          ;
965          ;
966          ;
967          ;
968          ;
969          ;
970          ;
971          ;
972          ;
973          ;
974          ;
975          ;
976          ;
977          ;
978          ;
979          ;
980          ;
981          ;
982          ;
983          ;
984          ;
985          ;
986          ;
987          ;
988          ;
989          ;
990          ;
991          ;
992          ;
993          ;
994          ;
995          ;
996          ;
997          ;
998          ;
999          ;
1000         ;

```

```
115 000230 103004 BCC 7$ :OK, CONTINUE.
116 000232 CALL ERR5 :INVALID-NUMERIC.
117 000236 000167 000420 JMP PR5X :EXIT.
118 :
119 :
120 :
121 :
122 000242 016767 000000G-000000G-7$: MOV BINWD,MEND :SET-UP-END-ADDRESS.
123 000250 CALL FIND :CHECK-FOR-LOOP-INDICATOR.
124 000254 103414 BCS 9$ :NO-LOOP.
125 000256 122711 000114 CMPB #'L,(R1) :CORRECT-INDICATOR.
126 000262 001404 BEQ 8$ :YES, CONTINUE.
127 000264 CALL ERR11 :LOOP-OPTION-ERROR.
128 000270 000167 000366 JMP PR5X
129 000274 8$:
130 000274 052767 000000G-000000G- BIS #OUT,HANG :SET-OUTPUT-CONTROL.
131 000302 CALL HANG :HOW-TO-STOP-LOOP.
132 :
133 000306 016701 000000G- 9$: MOV RTNPT,R1 :POINT-TO-ROUTINE.
134 000312 000171 000000 JMP @ (R1) :JUMP-TO-ROUTINE.
135 :
136 :
137 :
138 :
139 000316 012767 000004 000000G-PR5QX: MOV #S$QX, CODE :SET-MEMORY-SELECT-CODE = QEX.
140 000324 000420 BR PR5Q
141 :
142 :
143 :
144 000326 012767 000005 000000G-PR5QB: MOV #S$QB, CODE :SET-MEMORY-SELECT-CODE = SDLB PAGE.
145 000334 000414 BR PR5Q
146 :
147 :
148 :
149 000336 PR5QR: PRINT-QLB-REFERENCE-PAGE.
150 000336 012767 000006 000000G- MOV #S$QR, CODE :SET-MEMORY-SELECT-CODE = REFERENCE-PAGE.
151 000344 000410 BR PR5Q
152 :
153 :
154 :
155 000346 PR5SR: PRINT-SUBREAD-MEMORY.
156 000346 012767 000007 000000G- MOV #S$SR, CODE.
157 000354 016746 000000G- MOV SRHIGH, -(SP)
158 000360 016746 000000G- MOV SRLOW, -(SP) :SUPPLY-UPPER-MEMORY-LIMIT.
159 000362 000404 BR PR5GO :LOWER-LIMIT.
160 000366 PR5Q:
161 000366 016746 000000G- MOV SQHIGH, -(SP) :SUPPLY-UPPER-MEMORY-LIMIT.
162 000372 016746 000000G- MOV SQLOW, -(SP) :LOWER-LIMIT.
163 000376 PR5GO:
164 000376 CALL BUFSET :PREPARE-FOR-PRINT.
165 000402 103002 BCC 1$ :OK, CONTINUE.
166 000404 000167 000252 JMP PR5X :ERROR, EXIT.
167 :
168 000410 1$:
169 000410 CALL RDSM1
170 000414 016767 000000G-000000G- MOV DATA4, DATA1 :READ-16-BIT-MEMORY.
171 000422 CALL PRDATA :PRINT-IT.
```

SPPR- M 30-M1110 27-MAR-80 15:33 PAGE 5-3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

172.						
173	000426	066767	000000G.000000G.	ADD.	INCVAL,MSTR2.	:BUMP ADDRESS.
174	000434	026767	000000G.000000G.	CMP.	MSTR2,MEND.	:HAS UPPER MEMORY LIMIT BEEN REACHED.
175	000442	101762		BLOS.	1\$:NO, CONTINUE.
176	000444	032767	000000G.000000G.	BIT.	#LOOP,BASE.	:LOOP ON.
177	000452	001002		BNE.	8\$:YES, CONTINUE.
178	000454	000167	000202	JMP.	PR5X	
179	000460	016767	000000G.000000G.8\$:	MOV.	MSTR1,MSTR2.	:REINITIALIZE ADDRESS.
180	000466	000750		BR	1\$	

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
182. ; PRINT-SIDMEM (MULTIPLE PAGES)
183. ;
184. 000470 ; PR5S1::
185. 000470 012767 000010 000000G. MOV. #S$S1.CODE. ;SELECT-SIDMEM-1
186. 000476 000403 BR PR5S ;
187. 000500 ; PR5S2::
188. 000500 012767 000014 000000G. MOV. #S$S2.CODE. ;SELECT-SIDMEM-2
189. 000506 ; PR5S:
190. 000506 016746 000000G. MOV. SDHIGH, -(SP) ;SUPPLY UPPER MEMORY LIMIT
191. 000512 016746 000000G. MOV. SDLOW, -(SP) ;LOWER LIMIT
192. 000516 CALL. BUF93 ;PREPARE FOR READ
193. 000522 103002 BCC. 1$ ;OK, CONTINUE
194. 000524 000167 000132 JMP. PR5X ;ERROR, EXIT
195. ;
196. 000530 1$:
197. 000530 ;
198. 000534 ;
199. 000540 016767 000000G 000000G. CALL. RD5M1 ;READ MEMORY WORD A
200. 000546 005267 000000G. MOV. DATA4, DATA1 ;SAVE FOR PRINTING
201. 000552 INC. CODE ;SELECT WORD B
202. 000556 016767 000000G 000000G. CALL. RD5M1 ;READ MEMORY WORD B
203. 000564 005267 000000G. INC. CODE ;SAVE FOR PRINTING
204. 000570 CALL. RD5M1 ;SELECT WORD C
205. 000574 016767 000000G 000000G. MOV. DATA4, DATA3 ;READ MEMORY WORD C
206. 000602 RESTOR. CODE ;SAVE FOR PRINTING
207. ;
208. 000606 ;
209. 000612 103002 CALL. PRDATA. ;PRINT MEMORY CONTENTS
210. 000614 000167 000042 BCC. 20$ ;NOT END OF MEMORY
211. ; JMP. PR5X ;END OF MEMORY
212. 000620 20$:
213. 000620 066767 000000G 000000G. ADD. INCVAL, MSTR2. ;BUMP ADDRESS
214. 000626 026767 000000G 000000G. CMP. MSTR2, MEND. ;HAS UPPER MEMORY LIMIT BEEN REACHED
215. 000634 101735 BLOS. 1$ ;NO, CONTINUE
216. 000636 032767 000000G 000000G. BIT. #LOOP, BASE. ;LOOP ON
217. 000644 001002 BNE. 8$ ;YES, CONTINUE
218. 000646 000167 000010 JMP. PR5X
219. 000652 016767 000000G 000000G 8$: MOV. MSTR2, MSTR2. ;REINITIALIZE ADDRESS
220. 000660 000723 BR. 1$
221. ;
222. ;
223. 000662 ; PR5X:
224. 000662 012746 000000 MOV. #S$CLR, -(SP) ;CLEAR SP
225. 000666 CALL. SPCR ;WRITE TO CONTROL REG
226. 000672 042767 000000C 000000G. BIC. #<ONCE+OUT>, BASE. ;CLEAR PROMPT FLAG
227. 000700 CALL. KILL ;KILL AST (IF THERE WAS ONE)
228. 000704 000167 000000G. JMP. SPSXX
```

```
230 ;
231 ; PRINT SINGLE WORD MEMORY
232 ;
233 000710 RD5M1:
234 000710 012746 000001 MOV. #S$LA, -(SP) ; ADDRESS SELECT
235 000714 CALL SPCR ; SEND TO SP CONTROL REG
236 000720 016746 000000G MOV. MSTR2, -(SP) ; ACTUAL ADDRESS
237 000724 CALL LBSP ; SEND TO SP
238 000730 016746 000000G MOV. CODE, -(SP) ; SELECT MEMORY
239 000734 CALL SPCR
240 000740 CALL SPLB ; DO SP TO LOD BUS
241 000744 012667 000000G MOV. (SP)+, DATA4 ; FETCH DATA
242 000750 RETURN
243 ;
244 000001 .END
```

ALUCKE = 040000	BYTE39 = 000047	BYTE90 = 000132	MMADDR = 000100	Q\$ICP = 000006
ALUDE = 004000	BYTE4 = 000004	BYTE91 = 000133	MMLEFT = 000002	Q\$IHB = 000003
A01 = 010000	BYTE40 = 000050	BYTE92 = 000134	MMOE = 000004	Q\$IHRL = 000002
BASE = ***** GX	BYTE41 = 000051	BYTE93 = 000135	MMWRTE = 000010	Q\$IMRP = 000007
BINWD = ***** GX	BYTE42 = 000052	BYTE94 = 000136	MNOBRE = 100000	Q\$LBD = 001000
BITVAL = 000000	BYTE43 = 000053	BYTE95 = 000137	MREN1 = 000001	Q\$LBDF = 001001
BIT0 = 000001	BYTE44 = 000054	BYTE96 = 000140	MREN2 = 020000	Q\$LBP = 000001
BIT1 = 000002	BYTE45 = 000055	BYTE97 = 000141	MSTR1 = ***** GX	Q\$LDMD = 000004
BIT10 = 002000	BYTE46 = 000056	BYTE98 = 000142	MSTR2 = ***** GX	Q\$LDPP = 002000
BIT11 = 004000	BYTE47 = 000057	BYTE99 = 000143	MSYN = 000040	Q\$LHP = 010000
BIT12 = 010000	BYTE48 = 000060	BYTVAL = 000144	N = 000144	Q\$MNC = 140000
BIT13 = 020000	BYTE49 = 000061	CBKALL = 001000	ONCE = ***** GX	Q\$MR = 000052
BIT14 = 040000	BYTE5 = 000005	CBKCLK = 000400	OUT = ***** GX	Q\$MRP = 000040
BIT15 = 100000	BYTE50 = 000062	CNOBRE = 100000	PACK = ***** GX	Q\$MRP2 = 000240
BIT2 = 000004	BYTE51 = 000063	CODE = ***** GX	PLB = 000010	Q\$MSC = 040000
BIT3 = 000010	BYTE52 = 000064	CPCCEN = 010000	PLC = 000020	Q\$MSET = 000004
BIT4 = 000020	BYTE53 = 000065	CPREAD = 040000	PLD = 000030	Q\$MSP = 100000
BIT5 = 000040	BYTE54 = 000066	CPURTE = 020000	PLRWR = 000200	Q\$NCLK = 176000
BIT6 = 000100	BYTE55 = 000067	CSADDR = 000004	PLREN = 000200	Q\$PP = 000100
BIT7 = 000200	BYTE56 = 000070	CSEQCI = 100000	PRDATA = ***** GX	002: Q\$PPSW = 000320
BIT8 = 000400	BYTE57 = 000071	CSOE = 000040	PR5G0 = 000376R	Q\$PP2 = 000300
BIT9 = 001000	BYTE58 = 000072	CSURTE = 000100	PR5LN = 000030 G	002: Q\$QHLT = 000013
BUFSET = ***** GX	BYTE59 = 000073	DATA1 = ***** GX	PR5Q = 000366R	002: Q\$QL = 000043
BUFS3 = ***** GX	BYTE6 = 000006	DATA2 = ***** GX	PR5QB = 000326RG	002: Q\$QLA = 000053
BYTE0 = 000000	BYTE60 = 000074	DATA3 = ***** GX	PR5QR = 000336RG	002: Q\$QLB = 000054
BYTE1 = 000001	BYTE61 = 000075	DATA4 = ***** GX	PR5QX = 000316RG	002: Q\$QLR = 000001
BYTE10 = 000012	BYTE62 = 000076	DBR.RD = 000001	PR5S = 000506R	002: Q\$QW = 000042
BYTE11 = 000013	BYTE63 = 000077	DB\$CPP = 001457	PR5SR = 000346RG	002: Q\$RDCD = 000005
BYTE12 = 000014	BYTE64 = 000100	DB\$SPT = 000026	PR5S1 = 000470RG	002: Q\$RDMD = 000006
BYTE13 = 000015	BYTE65 = 000101	DB\$TPC = 000023	PR5S2 = 000500RG	002: Q\$REBK = 001000
BYTE14 = 000016	BYTE66 = 000102	DISPGS = 100000	PR5TBL = 000000RG	002: Q\$RNC = 006000
BYTE15 = 000017	BYTE67 = 000103	DIAAWR = 000005	PR5X = 000662R	Q\$RSC = 004000
BYTE16 = 000020	BYTE68 = 000104	DMARRD = 000003	QR\$CR1 = 176420	Q\$RSET = 000010
BYTE17 = 000021	BYTE69 = 000105	DMARWR = 000004	QR\$CR2 = 176422	Q\$SM = 100000
BYTE18 = 000022	BYTE7 = 000007	ENBR = 010000	QR\$LBR = 176424	Q\$SP = 000120
BYTE19 = 000023	BYTE70 = 000106	ERR11 = ***** GX	Q\$ATTN = 000100	Q\$SP2 = 000340
BYTE2 = 000002	BYTE71 = 000107	ERR4 = ***** GX	Q\$BCL = 000001	RD5M1 = 000710R 002
BYTE20 = 000024	BYTE72 = 000110	ERR5 = ***** GX	Q\$CCCP = 000040	RGQ.EN = 000200
BYTE21 = 000025	BYTE73 = 000111	ERR7 = ***** GX	Q\$CHB = 000400	RGQ.VA = 020000
BYTE22 = 000026	BYTE74 = 000112	FIND = ***** GX	Q\$CHRL = 000200	RTNPT = ***** GX
BYTE23 = 000027	BYTE75 = 000113	HANG = ***** GX	Q\$CLR = 000040	SCAN = ***** GX
BYTE24 = 000030	BYTE76 = 000114	INCVAL = ***** GX	Q\$CNC = 030000	SDHIGH = ***** GX
BYTE25 = 000031	BYTE77 = 000115	KILL = ***** GX	Q\$CP = 000060	SDLOW = ***** GX
BYTE26 = 000032	BYTE78 = 000116	LBSP = ***** GX	Q\$CPCD = 000010	SEQ.CI = 000010
BYTE27 = 000033	BYTE79 = 000117	LOC.EN = 000100	Q\$CP2 = 000260	SPCR = ***** GX
BYTE28 = 000034	BYTE8 = 000010	LOC.WA = 040000	Q\$CSC = 010000	SPLB = ***** GX
BYTE29 = 000035	BYTE80 = 000120	LOC.WB = 100000	Q\$CSEL = 000360	SPPRS = 000030RG 002
BYTE3 = 000003	BYTE81 = 000121	LOOP = ***** GX	Q\$CSET = 000002	SPSXX = ***** GX
BYTE30 = 000036	BYTE82 = 000122	MAREN1 = 000001	Q\$CHB = 020000	SOHIGH = ***** GX
BYTE31 = 000037	BYTE83 = 000123	MAREN2 = 004000	Q\$DMA = 000001	SQLOW = ***** GX
BYTE32 = 000040	BYTE84 = 000124	MARLOD = 010000	Q\$ENBK = 040000	SRHIGH = ***** GX
BYTE33 = 000041	BYTE85 = 000125	MAROUT = 000002	Q\$ENOP = 020000	SRLQW = ***** GX
BYTE34 = 000042	BYTE86 = 000126	MAR.LO = 002000	Q\$FAL = 004000	S\$CLR = ***** GX
BYTE35 = 000043	BYTE87 = 000127	MAR.OU = 000040	Q\$FC = 000045	S\$LA = 000001
BYTE36 = 000044	BYTE88 = 000130	MBKALL = 001000	Q\$FO = 000044	S\$QB = 000005
BYTE37 = 000045	BYTE89 = 000131	MBKCLK = 000400	Q\$FP = 000046	S\$QR = 000006
BYTE38 = 000046	BYTE9 = 000011	MEND = ***** GX	Q\$HBF = 000002	

SPPR: M 000-M1110 27-MAR-80 15:33 PAGE 7-2
SYMBOL TABLE

Approved For Release 2005/07/11 : CIA-RDP85-00514R000200020001-3

S#BX = 000004	T#IB = 000026	WORD15 = 000036	WORD44 = 000130	WORD73 = 000222
S#SR = 000007	T#IBAR = 000024	WORD16 = 000040	WORD45 = 000132	WORD74 = 000224
S#S1 = 000010	T#IBE = 020000	WORD17 = 000042	WORD46 = 000134	WORD75 = 000226
S#S2 = 000014	T#IBF = 040000	WORD18 = 000044	WORD47 = 000136	WORD76 = 000230
TD#CTR = 176370	T#ICD = 000040	WORD19 = 000046	WORD48 = 000140	WORD77 = 000232
TD#CTW = 176360	T#MODE = 004000	WORD2 = 000004	WORD49 = 000142	WORD78 = 000234
TD#INL = 004000	T#OB = 000036	WORD20 = 000050	WORD5 = 000012	WORD79 = 000236
TD#MEM = 000270	T#OBE = 004000	WORD21 = 000052	WORD50 = 000144	WORD8 = 000020
TD#OAR = 176344	T#OBF = 010000	WORD22 = 000054	WORD51 = 000146	WORD80 = 000240
TD#OTR = 176346	T#OBRA = 000034	WORD23 = 000056	WORD52 = 000150	WORD81 = 000242
TD#QRD = 000274	T#OBWA = 000032	WORD24 = 000060	WORD53 = 000152	WORD82 = 000244
TD#SW = 176376	T#OUTA = 100000	WORD25 = 000062	WORD54 = 000154	WORD83 = 000246
TD#TAR = 176372	T#RBD0 = 000200	WORD26 = 000064	WORD55 = 000156	WORD84 = 000250
TD#TAW = 176362	T#RNB = 000040	WORD27 = 000066	WORD56 = 000160	WORD85 = 000252
TD#TDR = 176374	T#RSET = 040000	WORD28 = 000070	WORD57 = 000162	WORD86 = 000254
TD#TDW = 176364	T#SC = 000022	WORD29 = 000072	WORD58 = 000164	WORD87 = 000256
T#AD = 000020	T#SCLK = 020000	WORD3 = 000006	WORD59 = 000166	WORD88 = 000260
T#BA = 000002	T#SEG1 = 000000	WORD30 = 000074	WORD6 = 000014	WORD89 = 000262
T#BD = 000010	T#SEG2 = 000001	WORD31 = 000076	WORD60 = 000170	WORD9 = 000022
T#BSO = 100000	T#SEG3 = 000002	WORD32 = 000100	WORD61 = 000172	WORD90 = 000264
T#BT = 000020	T#SO = 000001	WORD33 = 000102	WORD62 = 000174	WORD91 = 000266
T#BTAR = 000030	T#UBUS = 100000	WORD34 = 000104	WORD63 = 000176	WORD92 = 000270
T#BTD = 002000	T#1CLK = 000400	WORD35 = 000106	WORD64 = 000200	WORD93 = 000272
T#CD = 000100	T#GBEN = 000020	WORD36 = 000110	WORD65 = 000202	WORD94 = 000274
T#CLK = 002000	UBD, IN = 000020	WORD37 = 000112	WORD66 = 000204	WORD95 = 000276
T#DISK = 000200	WORD0 = 000000	WORD38 = 000114	WORD67 = 000206	WORD96 = 000300
T#DRD = 000004	WORD1 = 000002	WORD39 = 000116	WORD68 = 000210	WORD97 = 000302
T#EMEM = 010000	WORD10 = 000024	WORD4 = 000010	WORD69 = 000212	WORD98 = 000304
T#FSAA = 000000	WORD11 = 000026	WORD40 = 000120	WORD7 = 000016	WORD99 = 000306
T#FSAB = 000004	WORD12 = 000030	WORD41 = 000122	WORD70 = 000214	WORDVAL = 000310
T#FSAC = 000014	WORD13 = 000032	WORD42 = 000124	WORD71 = 000216	XTREAD = 001000
T#FSB2 = 000010	WORD14 = 000034	WORD43 = 000126	WORD72 = 000220	XTWRITE = 000400

. ABS. 000000 000
000000 001
SPPR: 000752 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3411 WORDS (14 PAGES)
DYNAMIC MEMORY: 4916 WORDS (18 PAGES)
ELAPSED TIME: 00:00:46
SPPR, SPPR/SP=C20,1JIM,C20,1JSPPR

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

1      .TITLE..SPREST...
2 000000      .PSECT: SPREST:
3
4
5
6      HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS
7      SUBDOCUMENT PROCESSOR TEST ROUTINES
8
9      COMMANDS:
10     PS      SWITCH QLB PAGES
11
12     ONCE A COMMAND HAS BEEN EXECUTED (OR AN ERROR ENCOUNTERED)
13     THIS MODULE RETURNS CONTROL TO THE MODULE SP AT LOCATION
14     'SPSXX'
15 000000      PS5::
16 000000      CALL      KILL
17 000004      000167      000000G      ;KILL AST
18      JMP      SPSXX      ;EXIT
19 000010      CL5::
20 000010      CALL      CL
21 000014      000167      000000G      ;CALL ROUTINE IN MAIN
22      JMP      SPSXX      ;EXIT
23      000001      .END

```


ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	Q\$LBR = 176424	Q\$SM = 100000
ALUDE = 004000	BYTE43 = 000053	BYTE95 = 000137	Q\$ATTN = 000100	Q\$SP = 000120
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	Q\$BCL = 000001	Q\$SP2 = 000340
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	Q\$CCCP = 000040	RGQ.EN = 000200
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	Q\$CHB = 000400	RGQ.VA = 020000
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	Q\$CHRL = 000200	SEQ.CI = 000010
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	Q\$CLR = 000040	SPSXX = ***** GX
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	Q\$CNC = 030000	S\$CLR = 000000
BIT12 = 010000	BYTE5 = 000005	CBKCLK = 000400	Q\$CP = 000060	S\$LA = 000001
BIT13 = 020000	BYTE50 = 000062	CL = ***** GX	Q\$CPCC = 000010	S\$OB = 000005
BIT14 = 040000	BYTE51 = 000063	CLS = 000010RG	Q\$CP2 = 000260	S\$OR = 000006
BIT15 = 100000	BYTE52 = 000064	CNOBRE = 100000	Q\$CSC = 010000	S\$QX = 000004
BIT2 = 000004	BYTE53 = 000065	CPCCEN = 010000	Q\$CSEL = 000360	S\$SR = 000007
BIT3 = 000010	BYTE54 = 000066	CPREAD = 040000	Q\$CSET = 000002	S\$S1 = 000010
BIT4 = 000020	BYTE55 = 000067	CPURTE = 020000	Q\$CSP = 020000	S\$S2 = 000014
BIT5 = 000040	BYTE56 = 000070	CSADRD = 000004	Q\$DMA = 000001	TD\$CTR = 176370
BIT6 = 000100	BYTE57 = 000071	CSEQCI = 100000	Q\$ENBK = 040000	TD\$CTW = 176360
BIT7 = 000200	BYTE58 = 000072	C\$OE = 000040	Q\$ENOP = 020000	TD\$INL = 004000
BIT8 = 000400	BYTE59 = 000073	CSURTE = 000100	Q\$FAL = 004000	TD\$MEM = 000270
BIT9 = 001000	BYTE6 = 000006	DBR.RD = 000001	Q\$FC = 000045	TD\$OAR = 176344
BYTE0 = 000000	BYTE60 = 000074	DB\$CPP = 001457	Q\$FO = 000044	TD\$OTR = 176346
BYTE1 = 000001	BYTE61 = 000075	DB\$SPT = 000026	Q\$FP = 000046	TD\$ORD = 000274
BYTE10 = 000012	BYTE62 = 000076	DB\$TPC = 000023	Q\$HBF = 000002	TD\$SW = 176376
BYTE11 = 000013	BYTE63 = 000077	DISPGS = 100000	Q\$ICP = 000006	TD\$STAR = 176372
BYTE12 = 000014	BYTE64 = 000100	DMAAUR = 000005	Q\$IHB = 000003	TD\$TAU = 176362
BYTE13 = 000015	BYTE65 = 000101	DMARRD = 000003	Q\$IHRL = 000002	TD\$TDR = 176374
BYTE14 = 000016	BYTE66 = 000102	DMARUR = 000004	Q\$IMRP = 000007	TD\$TDW = 176364
BYTE15 = 000017	BYTE67 = 000103	ENBR = 010000	Q\$LBD = 001000	T\$AD = 000020
BYTE16 = 000020	BYTE68 = 000104	KILL = ***** GX	Q\$LBDP = 001001	T\$BA = 000002
BYTE17 = 000021	BYTE69 = 000105	LOC.EN = 000100	Q\$LBP = 000001	T\$BD = 000010
BYTE18 = 000022	BYTE7 = 000007	LOC.WA = 040000	Q\$LCD = 000003	T\$BSO = 100000
BYTE19 = 000023	BYTE70 = 000106	LOC.WB = 100000	Q\$LMD = 000004	T\$BT = 000020
BYTE2 = 000002	BYTE71 = 000107	MAREN1 = 000001	Q\$LDPP = 002000	T\$BTAR = 000030
BYTE20 = 000024	BYTE72 = 000110	MAREN2 = 004000	Q\$LHP = 010000	T\$BTD = 002000
BYTE21 = 000025	BYTE73 = 000111	MARLOD = 010000	Q\$MNC = 140000	T\$CD = 000100
BYTE22 = 000026	BYTE74 = 000112	MAROUT = 000002	Q\$MR = 000052	T\$CLK = 002000
BYTE23 = 000027	BYTE75 = 000113	MAR.LO = 002000	Q\$MRP = 000040	T\$DISK = 000200
BYTE24 = 000030	BYTE76 = 000114	MAR.OU = 000040	Q\$MRP2 = 000240	T\$DRD = 000004
BYTE25 = 000031	BYTE77 = 000115	MBKALL = 001000	Q\$MSC = 040000	T\$EMEM = 010000
BYTE26 = 000032	BYTE78 = 000116	MBKCLK = 000400	Q\$MSET = 000004	T\$FSAA = 000000
BYTE27 = 000033	BYTE79 = 000117	MMADDR = 000100	Q\$MSP = 100000	T\$FSAB = 000004
BYTE28 = 000034	BYTE8 = 000010	MMLEFT = 000002	Q\$NCLK = 176000	T\$FSAC = 000014
BYTE29 = 000035	BYTE80 = 000120	MMOE = 000004	Q\$PP = 000100	T\$FSB2 = 000010
BYTE3 = 000003	BYTE81 = 000121	MMURTE = 000010	Q\$PPSW = 000320	T\$IB = 000026
BYTE30 = 000036	BYTE82 = 000122	MNOBRE = 100000	Q\$PP2 = 000300	T\$IBAR = 000024
BYTE31 = 000037	BYTE83 = 000123	MREN1 = 000001	Q\$QHLT = 000013	T\$IBE = 020000
BYTE32 = 000040	BYTE84 = 000124	MREN2 = 020000	Q\$QL = 000043	T\$IBF = 040000
BYTE33 = 000041	BYTE85 = 000125	MSYN = 000040	Q\$QLA = 000053	T\$ICD = 000040
BYTE34 = 000042	BYTE86 = 000126	N = 000144	Q\$QLB = 000054	T\$MODE = 004000
BYTE35 = 000043	BYTE87 = 000127	PLB = 000010	Q\$QLR = 000001	T\$OB = 000036
BYTE36 = 000044	BYTE88 = 000130	PLC = 000020	Q\$QW = 000042	T\$OBE = 004000
BYTE37 = 000045	BYTE89 = 000131	PLD = 000030	Q\$RDCD = 000005	T\$OBF = 010000
BYTE38 = 000046	BYTE9 = 000011	PLRWUR = 000200	Q\$RDMD = 000006	T\$OBRA = 000034
BYTE39 = 000047	BYTE90 = 000132	PLR.EN = 000200	Q\$REBK = 001000	T\$OUTH = 100000
BYTE4 = 000004	BYTE91 = 000133	PSS = 000000RG	Q\$RNC = 000000	T\$RBD = 000200
BYTE40 = 000050	BYTE92 = 000134	QR\$CR1 = 176420	Q\$RSC = 004000	T\$RNB = 000040
BYTE41 = 000051	BYTE93 = 000135	QR\$CP2 = 176423	Q\$RSET = 000010	

SPREST- M-RO-M1110 27-MAR-80 15:33 PAGE 5-2.
SYMBOL TABLE

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

T\$RSET= 040000	WORD2= 000004	WORD40= 000120	WORD61= 000172	WORD82= 000244
T\$SC= 000022	WORD20= 000050	WORD41= 000122	WORD62= 000174	WORD83= 000246
T\$SCLK= 020000	WORD21= 000052	WORD42= 000124	WORD63= 000176	WORD84= 000250
T\$SEG1= 000000	WORD22= 000054	WORD43= 000126	WORD64= 000200	WORD85= 000252
T\$SEG2= 000001	WORD23= 000056	WORD44= 000130	WORD65= 000202	WORD86= 000254
T\$SEG3= 000002	WORD24= 000060	WORD45= 000132	WORD66= 000204	WORD87= 000256
T\$SO= 000001	WORD25= 000062	WORD46= 000134	WORD67= 000206	WORD88= 000260
T\$UBUS= 100000	WORD26= 000064	WORD47= 000136	WORD68= 000210	WORD89= 000262
T\$1CLK= 000400	WORD27= 000066	WORD48= 000140	WORD69= 000212	WORD9= 000022
T\$BEN= 000020	WORD28= 000070	WORD49= 000142	WORD7= 000016	WORD90= 000264
UBD, IN= 000020	WORD29= 000072	WORD5= 000012	WORD70= 000214	WORD91= 000266
WORD0= 000000	WORD3= 000006	WORD50= 000144	WORD71= 000216	WORD92= 000270
WORD1= 000002	WORD30= 000074	WORD51= 000146	WORD72= 000220	WORD93= 000272
WORD10= 000024	WORD31= 000076	WORD52= 000150	WORD73= 000222	WORD94= 000274
WORD11= 000026	WORD32= 000100	WORD53= 000152	WORD74= 000224	WORD95= 000276
WORD12= 000030	WORD33= 000102	WORD54= 000154	WORD75= 000226	WORD96= 000300
WORD13= 000032	WORD34= 000104	WORD55= 000156	WORD76= 000230	WORD97= 000302
WORD14= 000034	WORD35= 000106	WORD56= 000160	WORD77= 000232	WORD98= 000304
WORD15= 000036	WORD36= 000110	WORD57= 000162	WORD78= 000234	WORD99= 000306
WORD16= 000040	WORD37= 000112	WORD58= 000164	WORD79= 000236	WORDVAL= 000310
WORD17= 000042	WORD38= 000114	WORD59= 000166	WORD8= 000020	XTREAD= 001000
WORD18= 000044	WORD39= 000116	WORD6= 000014	WORD80= 000240	XTWRITE= 000400
WORD19= 000046	WORD4= 000010	WORD60= 000170	WORD81= 000242	

. ABS. 000000 000
000000 001
SPREST 000020 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3001 WORDS (12 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:39
SPREST,SPREST/SP=[20,1]IM,[20,1]SPREST

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```

1      .TITLE--SPSUB....
2 000000 .PSECT: SPSUB.
3      ;
4      ;
5      .MCALL: WTSE$S,CLEF$S.
6      000003 EFN.3 = 3
7      ;
8      ;
9      ;
10     ;
11     ;
12     ;
13     ;
14     ;
15     ;
16     ;
17     ;
18     ;
19     ;
20 000000 LBSP::
21 000000 016667 000002 176424
22 000006 012746 001001
23 000012 052716 000360
24 000016 012746 176000
25 000022 052716 000340
26 000026
27     ;
28 000032 012746 006000
29 000036 012746 001000
30 000042
31     ;
32     ;
33     ;
34 000046 012746 001001
35 000052 052716 000360
36 000056 012746 176000
37 000062
38     ;
39 000066 011666 000002
40 000072 005726
41 000074

      MOV. 2(SP),Q$LBR. ;MOVE DATA TO LOD-BUS-REG.
      MOV. #<Q$LBD+Q$LBP>,-(SP) ;CLR DRIVE AND PULSE
      BIS. #<Q$CSEL>,(SP) ;CLR SELECTION BITS
      MOV. #<Q$NCLK>,-(SP) ;SET NO-CLOCKS
      BIS. #Q$SP2,(SP) ;SELECT SP
      CALL CSR1 ;
      ;
      MOV. #Q$RNC,-(SP) ;CLEAR CP NO-CLOCK BITS
      MOV. #Q$LBD,-(SP) ;SET LOD-BUS-DRIVE
      CALL CSR1 ;
      ;
      DE-SELECTION
      ;
      MOV. #<Q$LBD+Q$LBP>,-(SP) ;CLEAR DRIVE AND PULSE
      BIS. #<Q$CSEL>,(SP) ;CLR SELECTION BITS
      MOV. #<Q$NCLK>,-(SP) ;SET NO-CLOCKS
      CALL CSR1 ;
      ;
      MOV. (SP),2(SP) ;MOVE RETURN ADDRESS DOWN STACK
      TST. (SP)+ ;POINT TO RETURN ADDRESS
      RETURN.

```

```

43      ;
44      ;
45      ; DATA TRANSFER
46      ; LOD-BUS REGISTER TO A DESTINATION ON THE SP-BUS
47      ; SINGLE-CLOCK SEQUENCER ONLY
48      ;
49      ; INPUT:
50      ; 2(SP) DATA FOR PRE-SELECTED SP DESTINATION
51      ;
52      ;
53 000076      ; LBSSC::
54 000076 016667 000002 176424 MOV 2(SP),Q$LBR ; MOVE DATA TO LOD-BUS-REG
55 000104 012746 001001 MOV #<Q$LBD+Q$LBP>,-(SP) ; CLEAR DRIVE AND PULSE
56 000110 052716 000360 BIS #Q$CSEL,(SP) ; CLR SELECTION BITS
57 000114 012746 176000 MOV #<Q$NCLK>,-(SP) ; SET NO-CLOCKS
58 000120 052716 000340 BIS #Q$SP2,(SP) ; SELECT SP
59 000124      ; CALL CSR1 ; WRITE CONTROL REGISTER
60      ;
61 000130 012746 005000 MOV #Q$RNC,-(SP) ; CLEAR SP NO-CLOCK BITS
62 000134 012746 005000 MOV #<Q$RSC+Q$LBD>,-(SP) ; SET SP-CLOCK
63 000140      ; CALL CSR1 ;
64      ;
65      ; DE-SELECTION
66      ;
67 000144 012746 001001 MOV #<Q$LBD+Q$LBP>,-(SP) ; CLEAR DRIVE AND PULSE
68 000150 052716 000360 BIS #Q$CSEL,(SP) ; CLR SELECTION BITS
69 000154 012746 176000 MOV #<Q$NCLK>,-(SP) ; SET NO-CLOCKS
70 000160      ; CALL CSR1 ;
71      ;
72 000164 011666 000002 MOV (SP),2(SP) ; MOVE RETURN ADDRESS DOWN STACK
73 000170 005726      TST (SP)+ ; POINT TO RETURN ADDRESS
74 000172      ; RETURN

```

```
76      ;
77      ;
78      ; DATA TRANSFER TO LOD BUS REG FROM SP.
79      ;
80      ;
81      ; OUTPUT:
82      ; (SP) DATA FROM PRE-SELECTED SP SOURCE.
83      ;
84      ;
85      ; SPLB::
86      000174 012746 001001 MOV. #<Q$LBD+Q$LBP>,-(SP) ;CLEAR DRIVE AND PULSE.
87      000200 052716 176360 BIS. #<Q$CSEL+Q$NCLK>,(SP) ;CLR SELECTION BITS/NO CLOCKS.
88      000204 012746 000340 MOV. #<Q$SP2>,-(SP) ;SET NO-CLOCKS.
89      000210 CALL CSR1 ;
90      000214 011646 MOV. (SP),-(SP) ;MOVE RETURN ADDR UP STACK.
91      000216 016766 176424 000002 MOV. Q$LBR,2(SP) ;MOVE DATA ONTO STACK.
92      000224 012746 000340 MOV. #Q$SP2,-(SP) ;CLEAR SP SELECT.
93      000230 012746 176000 MOV. #<Q$NCLK>,-(SP) ;SET NO-CLOCKS.
94      000234 CALL CSR1 ;
95      000240 RETURN ;
```

```
97      ;
98      ;
99      ;      SP CONTROL REGISTER LOADING.
100     ;
101     ;      INPUT:
102     ;      2(SP)   BIT SETTING FOR SP CONTROL REGISTER.
103     ;
104     ;
105     000242      SPCR::
106     000242      016667  000002  176424      MOV.      2(SP),QR$LBR.      ;CONTROL BITS DESTINED FOR SP.
107     000250      012746  001001      MOV.      *(<Q$LBD+Q$LBP>),-(SP) ;CLEAR DRIVE AND PULSE.
108     000254      052716  000360      BIS.      #Q$CSEL,(SP)      ;CLR SELECTION BITS.
109     000260      012746  000120      MOV.      #Q$SP, -(SP)      ;SELECT SP.
110     000264      CALL      CSR1
111     ;
112     000270      005046      CLR.      -(SP)      ;CLEAR NOTHING.
113     000272      012746  000001      MOV.      #Q$LBP, -(SP)      ;SET PULSE.
114     000276      CALL      CSR1
115     ;
116     000302      012746  000121      MOV.      *(<Q$SP+Q$LBP>),-(SP) ;CLEAR CR SELECTION AND PULSE.
117     000306      005046      CLR.      -(SP)      ;SET NOTHING.
118     000310      CALL      CSR1
119     ;
120     000314      011666  000002      MOV.      (SP),2(SP)      ;MOVE RETURN ADDRESS DOWN STACK.
121     000320      005726      TST.      (SP)+      ;POINT TO RETURN ADDRESS.
122     000322      RETURN.
123     ;
124     000001      .END.
```

ALUCKE=-.040000	BYTE42=-.000052	BYTE94=-.000136	QR\$LBR=-.176424	Q\$SM =-.100000
ALUOE=-.004000	BYTE43=-.000053	BYTE95=-.000137	Q\$ATTN=-.000100	Q\$SP =-.000120
A01 =-.010000	BYTE44=-.000054	BYTE96=-.000140	Q\$BCL =-.000001	Q\$SP2=-.000340
BITVAL=-.000000	BYTE45=-.000055	BYTE97=-.000141	Q\$CCCP=-.000040	RGQ.EN=-.000200
BIT0 =-.000001	BYTE46=-.000056	BYTE98=-.000142	Q\$CHB =-.000400	RGQ.VA=-.020000
BIT1 =-.000002	BYTE47=-.000057	BYTE99=-.000143	Q\$CHRL=-.000200	SEQ.CI=-.000010
BIT10=-.002000	BYTE48=-.000060	BYTVAL=-.000144	Q\$CLR =-.000040	SPCR 000242RG. 002
BIT11=-.004000	BYTE49=-.000061	CBKALL=-.001000	Q\$CNC =-.030000	SPLB 000174RG. 002
BIT12=-.010000	BYTE5 =-.000005	CBKCLK=-.000400	Q\$CP =-.000060	S\$CLR =-.000000
BIT13=-.020000	BYTE50=-.000062	CNOBRE=-.100000	Q\$CPCC=-.000010	S\$LA =-.000001
BIT14=-.040000	BYTE51=-.000063	CPCCEN=-.010000	Q\$CP2 =-.000260	S\$QB =-.000005
BIT15=-.100000	BYTE52=-.000064	CPREAD=-.040000	Q\$CSC =-.010000	S\$QR =-.000006
BIT2 =-.000004	BYTE53=-.000065	CPURTE=-.020000	Q\$CSEL=-.000360	S\$QX =-.000004
BIT3 =-.000010	BYTE54=-.000066	CSADRD=-.000004	Q\$CSET=-.000002	S\$SR =-.000002
BIT4 =-.000020	BYTE55=-.000067	CSEQCI=-.100000	Q\$CSP =-.020000	S\$S1 =-.000010
BIT5 =-.000040	BYTE56=-.000070	CSOE =-.000040	Q\$DMA =-.000001	S\$S2 =-.000014
BIT6 =-.000100	BYTE57=-.000071	CSR1 =-***** GX	Q\$ENBK=-.040000	TD\$CTR=-.176370
BIT7 =-.000200	BYTE58=-.000072	CSURTE=-.000100	Q\$ENOP=-.020000	TD\$CTN=-.176360
BIT8 =-.000400	BYTE59=-.000073	CSURTE=-.000100		TD\$INL=-.024270
BIT9 =-.000600	BYTE60=-.000074	CSURTE=-.000100		TD\$MEM=-.024270
BIT10=-.000800	BYTE61=-.000075	CSURTE=-.000100		TD\$QAR=-.176344
BIT11=-.001000	BYTE62=-.000076	CSURTE=-.000100		TD\$QTR=-.176345
BIT12=-.001200	BYTE63=-.000077	CSURTE=-.000100		TD\$QRD=-.233274
BIT13=-.001400	BYTE64=-.000078	CSURTE=-.000100		TD\$QSH=-.176376
BIT14=-.001600	BYTE65=-.000079	CSURTE=-.000100		
BIT15=-.001800	BYTE66=-.000080	CSURTE=-.000100		
BIT16=-.002000	BYTE67=-.000081	CSURTE=-.000100		
BIT17=-.002200	BYTE68=-.000082	CSURTE=-.000100		
BIT18=-.002400	BYTE69=-.000083	CSURTE=-.000100		
BIT19=-.002600	BYTE70=-.000084	CSURTE=-.000100		
BIT20=-.002800	BYTE71=-.000085	CSURTE=-.000100		
BIT21=-.003000	BYTE72=-.000086	CSURTE=-.000100		
BIT22=-.003200	BYTE73=-.000087	CSURTE=-.000100		
BIT23=-.003400	BYTE74=-.000088	CSURTE=-.000100		
BIT24=-.003600	BYTE75=-.000089	CSURTE=-.000100		
BIT25=-.003800	BYTE76=-.000090	CSURTE=-.000100		
BIT26=-.004000	BYTE77=-.000091	CSURTE=-.000100		
BIT27=-.004200	BYTE78=-.000092	CSURTE=-.000100		
BIT28=-.004400	BYTE79=-.000093	CSURTE=-.000100		
BIT29=-.004600	BYTE80=-.000094	CSURTE=-.000100		
BIT30=-.004800	BYTE81=-.000095	CSURTE=-.000100		
BIT31=-.005000	BYTE82=-.000096	CSURTE=-.000100		
BIT32=-.005200	BYTE83=-.000097	CSURTE=-.000100		
BIT33=-.005400	BYTE84=-.000098	CSURTE=-.000100		
BIT34=-.005600	BYTE85=-.000099	CSURTE=-.000100		
BIT35=-.005800	BYTE86=-.000100	CSURTE=-.000100		
BIT36=-.006000	BYTE87=-.000101	CSURTE=-.000100		
BIT37=-.006200	BYTE88=-.000102	CSURTE=-.000100		
BIT38=-.006400	BYTE89=-.000103	CSURTE=-.000100		
BIT39=-.006600	BYTE90=-.000104	CSURTE=-.000100		
BIT40=-.006800	BYTE91=-.000105	CSURTE=-.000100		
BIT41=-.007000	BYTE92=-.000106	CSURTE=-.000100		
BIT42=-.007200	BYTE93=-.000107	CSURTE=-.000100		
BIT43=-.007400	BYTE94=-.000108	CSURTE=-.000100		
BIT44=-.007600	BYTE95=-.000109	CSURTE=-.000100		
BIT45=-.007800	BYTE96=-.000110	CSURTE=-.000100		
BIT46=-.008000	BYTE97=-.000111	CSURTE=-.000100		
BIT47=-.008200	BYTE98=-.000112	CSURTE=-.000100		
BIT48=-.008400	BYTE99=-.000113	CSURTE=-.000100		
BIT49=-.008600	BYTVAL=-.000114	CSURTE=-.000100		
BIT50=-.008800	CBKALL=-.001000	CSURTE=-.000100		
BIT51=-.009000	CBKCLK=-.000400	CSURTE=-.000100		
BIT52=-.009200	CNOBRE=-.100000	CSURTE=-.000100		
BIT53=-.009400	CPCCEN=-.010000	CSURTE=-.000100		
BIT54=-.009600	CPREAD=-.040000	CSURTE=-.000100		
BIT55=-.009800	CPURTE=-.020000	CSURTE=-.000100		
BIT56=-.010000	CSADRD=-.000004	CSURTE=-.000100		
BIT57=-.010200	CSEQCI=-.100000	CSURTE=-.000100		
BIT58=-.010400	CSOE =-.000040	CSURTE=-.000100		
BIT59=-.010600	CSR1 =-***** GX	CSURTE=-.000100		
BIT60=-.010800	CSURTE=-.000100	CSURTE=-.000100		
BIT61=-.011000	CSURTE=-.000100	CSURTE=-.000100		
BIT62=-.011200	CSURTE=-.000100	CSURTE=-.000100		
BIT63=-.011400	CSURTE=-.000100	CSURTE=-.000100		
BIT64=-.011600	CSURTE=-.000100	CSURTE=-.000100		
BIT65=-.011800	CSURTE=-.000100	CSURTE=-.000100		
BIT66=-.012000	CSURTE=-.000100	CSURTE=-.000100		
BIT67=-.012200	CSURTE=-.000100	CSURTE=-.000100		
BIT68=-.012400	CSURTE=-.000100	CSURTE=-.000100		
BIT69=-.012600	CSURTE=-.000100	CSURTE=-.000100		
BIT70=-.012800	CSURTE=-.000100	CSURTE=-.000100		
BIT71=-.013000	CSURTE=-.000100	CSURTE=-.000100		
BIT72=-.013200	CSURTE=-.000100	CSURTE=-.000100		
BIT73=-.013400	CSURTE=-.000100	CSURTE=-.000100		
BIT74=-.013600	CSURTE=-.000100	CSURTE=-.000100		
BIT75=-.013800	CSURTE=-.000100	CSURTE=-.000100		
BIT76=-.014000	CSURTE=-.000100	CSURTE=-.000100		
BIT77=-.014200	CSURTE=-.000100	CSURTE=-.000100		
BIT78=-.014400	CSURTE=-.000100	CSURTE=-.000100		
BIT79=-.014600	CSURTE=-.000100	CSURTE=-.000100		
BIT80=-.014800	CSURTE=-.000100	CSURTE=-.000100		
BIT81=-.015000	CSURTE=-.000100	CSURTE=-.000100		
BIT82=-.015200	CSURTE=-.000100	CSURTE=-.000100		
BIT83=-.015400	CSURTE=-.000100	CSURTE=-.000100		
BIT84=-.015600	CSURTE=-.000100	CSURTE=-.000100		
BIT85=-.015800	CSURTE=-.000100	CSURTE=-.000100		
BIT86=-.016000	CSURTE=-.000100	CSURTE=-.000100		
BIT87=-.016200	CSURTE=-.000100	CSURTE=-.000100		
BIT88=-.016400	CSURTE=-.000100	CSURTE=-.000100		
BIT89=-.016600	CSURTE=-.000100	CSURTE=-.000100		
BIT90=-.016800	CSURTE=-.000100	CSURTE=-.000100		
BIT91=-.017000	CSURTE=-.000100	CSURTE=-.000100		
BIT92=-.017200	CSURTE=-.000100	CSURTE=-.000100		
BIT93=-.017400	CSURTE=-.000100	CSURTE=-.000100		
BIT94=-.017600	CSURTE=-.000100	CSURTE=-.000100		
BIT95=-.017800	CSURTE=-.000100	CSURTE=-.000100		
BIT96=-.018000	CSURTE=-.000100	CSURTE=-.000100		
BIT97=-.018200	CSURTE=-.000100	CSURTE=-.000100		
BIT98=-.018400	CSURTE=-.000100	CSURTE=-.000100		
BIT99=-.018600	CSURTE=-.000100	CSURTE=-.000100		
BIT100=-.018800	CSURTE=-.000100	CSURTE=-.000100		

BYTE12= 000014
BYTE13= 000015
BYTE14= 000016
BYTE15= 000017
BYTE16= 000020
BYTE17= 000021
BYTE18= 000022
BYTE19= 000023
BYTE20= 000024
BYTE21= 000025
BYTE22= 000026
BYTE23= 000027
BYTE24= 000030
BYTE25= 000031
BYTE26= 000032
BYTE27= 000033
BYTE28= 000034
BYTE29= 000035
BYTE30= 000036
BYTE31= 000037
BYTE32= 000040
BYTE33= 000041
BYTE34= 000042
BYTE35= 000043
BYTE36= 000044
BYTE37= 000045
BYTE38= 000046
BYTE39= 000047
BYTE40= 000050
BYTE41= 000051

BYTE64= 000100
BYTE65= 000102
BYTE66= 000103
BYTE67= 000103
BYTE68= 000104
BYTE69= 000105
BYTE70= 000106
BYTE71= 000107
BYTE72= 000110
BYTE73= 000111
BYTE74= 000112
BYTE75= 000113
BYTE76= 000114
BYTE77= 000115
BYTE78= 000116
BYTE79= 000117
BYTE80= 000120
BYTE81= 000121
BYTE82= 000122
BYTE83= 000123
BYTE84= 000124
BYTE85= 000125
BYTE86= 000126
BYTE87= 000127
BYTE88= 000130
BYTE89= 000131
BYTE90= 000132
BYTE91= 000133
BYTE92= 000134
BYTE93= 000135

DMARRD= 000003
EFN.3 = 000003
ENBR = 010000
LBSP = 000000RG
LBSSC = 000076RG
LOC.EN = 000100
LOC.WA = 040000
LOC.WB = 100000
MAREN1 = 000001
MAREN2 = 004000
MARLOD = 010000
MAROUT = 000002
MAR.LO = 002000
MAR.OU = 000040
MBKALL = 001000
MBKCLK = 000400
MMADRD = 000100
MMLEFT = 000002
MMOE = 000004
MMWRTE = 000010
MNOBRE = 100000
MREN1 = 000001
MREN2 = 020000
MSYN = 000040
N = 000144
PLB = 000010
PLC = 000020
PLD = 000030
PLRWR = 000200
PLR.EN = 000200
QR#CR1 = 176420
QR#CR2 = 176422

Q\$IRP = 000003
Q\$IRRC = 000002
Q\$IMRP = 000007
Q\$LBD = 001000
Q\$LBDP = 001001
Q\$LB = 000001
Q\$LCD = 000003
Q\$LDMD = 000004
Q\$LDPP = 002000
Q\$LHP = 010000
Q\$MNC = 140000
Q\$MR = 000052
Q\$MRP = 000040
Q\$MRP2 = 000240
Q\$MSC = 040000
Q\$MSET = 000004
Q\$MSP = 100000
Q\$NCLK = 176000
Q\$PP = 000100
Q\$PPSW = 000320
Q\$PP2 = 000300
Q\$QHLT = 000013
Q\$QL = 000043
Q\$QLA = 000053
Q\$QLB = 000054
Q\$QLR = 000001
Q\$QW = 000042
Q\$RD = 000005
Q\$RDMD = 000006
Q\$REBK = 001000
Q\$RNC = 006000
Q\$RSC = 004000
Q\$RSET = 000010

T\$AR = 176372
T\$TAJ = 176362
T\$TDR = 176374
T\$TDW = 176364
T\$AD = 000020
T\$BA = 000002
T\$BD = 000010
T\$BSO = 100000
T\$BT = 000020
T\$BTAR = 000030
T\$BTD = 002000
T\$CD = 000100
T\$CLK = 002000
T\$DISK = 000200
T\$DPD = 000004
T\$EMEM = 010000
T\$FSAA = 000000
T\$FSAB = 000004
T\$FSAC = 000014
T\$FSB2 = 000010
T\$IB = 000026
T\$IBAR = 000024
T\$IBE = 020000
T\$IBF = 040000
T\$ICD = 000040
T\$MODE = 004000
T\$OB = 000036
T\$OBE = 004000
T\$OBF = 010000
T\$OBRA = 000034
T\$OBWA = 000032
T\$OUTA = 100000
T\$RBD0 = 000200

SPSUB:..M1110 27-MAR-80 15:34 PAGE 8-2.
SYMBOL TABLE

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

T#RNB = 000040	WORD19 = 000046	WORD4 = 000010	WORD60 = 000170	WORD81 = 000242
T#RSET = 040000	WORD2 = 000004	WORD40 = 000120	WORD61 = 000172	WORD82 = 000244
T#SC = 000022	WORD20 = 000050	WORD41 = 000122	WORD62 = 000174	WORD83 = 000246
T#SCLK = 020000	WORD21 = 000052	WORD42 = 000124	WORD63 = 000176	WORD84 = 000250
T#SEG1 = 000000	WORD22 = 000054	WORD43 = 000126	WORD64 = 000200	WORD85 = 000252
T#SEG2 = 000001	WORD23 = 000056	WORD44 = 000130	WORD65 = 000202	WORD86 = 000254
T#SEG3 = 000002	WORD24 = 000060	WORD45 = 000132	WORD66 = 000204	WORD87 = 000256
T#SO = 000001	WORD25 = 000062	WORD46 = 000134	WORD67 = 000206	WORD88 = 000260
T#UBUS = 100000	WORD26 = 000064	WORD47 = 000136	WORD68 = 000210	WORD89 = 000262
T#1CLK = 000400	WORD27 = 000066	WORD48 = 000140	WORD69 = 000212	WORD9 = 000022
T#BBEN = 000020	WORD28 = 000070	WORD49 = 000142	WORD7 = 000016	WORD90 = 000264
UBD, IN = 000020	WORD29 = 000072	WORD5 = 000012	WORD70 = 000214	WORD91 = 000266
WORD0 = 000000	WORD3 = 000006	WORD50 = 000144	WORD71 = 000216	WORD92 = 000270
WORD1 = 000002	WORD30 = 000074	WORD51 = 000146	WORD72 = 000220	WORD93 = 000272
WORD10 = 000024	WORD31 = 000076	WORD52 = 000150	WORD73 = 000222	WORD94 = 000274
WORD11 = 000026	WORD32 = 000100	WORD53 = 000152	WORD74 = 000224	WORD95 = 000276
WORD12 = 000030	WORD33 = 000102	WORD54 = 000154	WORD75 = 000226	WORD96 = 000300
WORD13 = 000032	WORD34 = 000104	WORD55 = 000156	WORD76 = 000230	WORD97 = 000302
WORD14 = 000034	WORD35 = 000106	WORD56 = 000160	WORD77 = 000232	WORD98 = 000304
WORD15 = 000036	WORD36 = 000110	WORD57 = 000162	WORD78 = 000234	WORD99 = 000306
WORD16 = 000040	WORD37 = 000112	WORD58 = 000164	WORD79 = 000236	WORDVAL = 000310
WORD17 = 000042	WORD38 = 000114	WORD59 = 000166	WORD8 = 000020	XTREAD = 001000
WORD18 = 000044	WORD39 = 000116	WORD6 = 000014	WORD80 = 000240	XTURTE = 000400

. ABS. 000000 000
000000 001
SPSUB. 000324 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3112 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:42
SPSUB, SPSUB--SP=[20,1]IM,[20,1]SPSUB

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

AIDQR.TSK MEMORY ALLOCATION MAP TKB
27-MAR-80

PAGE 1

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

TASK. NAME. : AIDQR.
PARTITION NAME : GEN
IDENTIFICATION : 08
TASK. UIC. : [20.3].
STACK. LIMITS: 040176 041175 001000 00512.
PRG. XFR. ADDRESS: 043372.
TASK. ATTRIBUTES: AL,CP.
TOTAL ADDRESS WINDOWS: 1.
TASK. IMAGE. SIZE : 7744. WORDS.
TASK. ADDRESS. LIMITS: 040000 076167
R-W DISK BLK. LIMITS: 000042 000137 000076 00062.

AIDQR.TSK:7 OVERLAY DESCRIPTION:

BASE.	TOP.	LENGTH.	
----	----	-----	
040000	072013	032014	13324.
072014	072363	000350	00232.
072364	073557	001174	00636.
072364	073473	001110	00584.
072364	073657	001274	00700.
072364	073353	000770	00504.
072014	073517	001504	00836.
073520	074373	000654	00428.
073520	076167	002450	01320.
073520	075327	001610	00904.
073520	075407	001670	00952.
073520	075303	001564	00884.
072014	072647	000634	00412.
072014	072567	000554	00364.
072570	075233	002444	01316.
072570	075167	002400	01280.
072570	074167	001400	00768.
072014	072573	000560	00368.
072574	073647	001054	00556.
072574	073547	000754	00492.
072574	07261.	000020	00016.

QMAIN.
MRP.
MRLD.
MRPR.
MRREST.
MRBUG.
CP.
CPBUG1
CPBUG2.
CPREST.
CPLD.
CPPR.
BCE.
PPS.
PPLD.
PPPR.
PPREST.
SP.
SPLD.
SPPR.
SPREST

AIDOR.TSK:7 MEMORY ALLOCATION MAP TKB
QMAIN: 27-MAR-80

PAGE 2

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

*** ROOT SEGMENT: QMAIN.

R/W MEM LIMITS: 040000 072013 032014 13324.
DISK BLK LIMITS: 000042 000074 000033 00027.

MEMORY ALLOCATION SYNOPSIS:

SECTION	TITLE	IDENT	FILE
BLK: (RW, I, LCL, REL, CON)	041176	011660	05040.
CPSUB: (RW, I, LCL, REL, CON)	053056	000502	00322.
MRPSUB: (RW, I, LCL, REL, CON)	053560	000506	00326.
PPSUB: (RW, I, LCL, REL, CON)	054266	000766	00502.
\$\$ALER: (RW, I, LCL, REL, CON)	055254	000024	00020.
\$\$ALVC: (RW, D, LCL, REL, CON)	055300	000060	00048.
\$\$AUTO: (RW, I, LCL, REL, CON)	055360	000130	00008.
\$\$FSR1: (RW, D, GBL, REL, OVR)	055510	001020	00528.
\$\$FSR2: (RW, D, GBL, REL, CON)	056530	000104	00068.
\$\$MRKS: (RO, I, LCL, REL, OVR)	071544	000076	00062.
\$\$OVD: (RW, D, LCL, REL, OVR)	056634	000020	00016.
\$\$OVR: (RW, I, LCL, ABS, CON)	000000	000000	00000.
\$\$RDSG: (RO, I, LCL, REL, OVR)	071642	000150	00104.
\$\$RESL: (RW, I, LCL, REL, CON)	056654	012270	05304.
\$\$RGDS: (RW, D, LCL, REL, CON)	071144	000000	00000.
\$\$RTS: (RW, I, GBL, REL, OVR)	071144	000002	00002.
\$\$SGD0: (RW, D, LCL, REL, OVR)	071146	000000	00000.
\$\$SGD1: (RW, D, LCL, REL, CON)	071146	000374	00252.
\$\$SGD2: (RW, D, LCL, REL, OVR)	071542	000002	00002.
\$\$UNDS: (RW, D, LCL, REL, CON)	071544	000000	00000.

GLOBAL SYMBOLS:

APLACE	041234-R	BUFS3	043754-R	CSR1	043606-R	ENFILE	046026-R	ERWORD	041226-R	HLOW	041452-R	LHLOW	041446-R
ASCIZ	043063-R	BUFS4	043736-R	DATA1	041404-R	ERR1	046122-R	FAHIGH	041440-R	INCVAL	041376-R	LOOP	000004
ASTFLG	000200	CDHIGH	041430-R	DATA2	041406-R	ERR10	046056-R	FALOW	041442-R	IO.DET	002000	LOOPR	044602-R
ASTURD	041364-R	CDLOW	041432-R	DATA3	041410-R	ERR11	046052-R	FIND	045206-R	IO.RVB	010400	LUN.TT	000001
BASE	041232-R	CL	043646-R	DATA4	041412-R	ERR12	046046-R	GCMBLK	043064-R	IO.UVB	011000	MDHIGH	041420-R
CE	055320-R	COMXX	043532-R	EFBUF	041212-R	ERR2	046116-R	GCMBUF	041236-R	KILL	044762-R	MDLOW	041422-R
JINWD	041230-R	CONSOL	045634-R	EFN.1	000001	ERR3	046112-R	GCMLEN	041360-R	LBCP	053224-R	MEND	041374-R
JLHIGH	041454-R	CP	055310-R	EFN.2	000002	ERR4	046106-R	GCMPT	041362-R	LBCSC	053322-R	MMHIGH	041414-R
JLLOW	041456-R	CPCR	053470-R	EFN.3	000003	ERR5	046102-R	GCONLY	044452-R	LBMRP	053732-R	MMLOW	041416-R
BREAK	004000	CPCRA	053476-R	EFN.33	000041	ERR6	046076-R	G.DPRM	000160	LBMSC	054030-R	MRP	055300-R
BUSET	044010-R	CPLB	053420-R	EFN.4	000004	ERR7	046072-R	HANG	044624-R	LBPB	054266-R	MRPCR	054176-R
BUFMS	044026-R	CSHIGH	041424-R	ENDMEM	046032-R	ERR8	046066-R	HANG2	044710-R	LBPSC	054364-R	MRPCRA	054204-R
BUFS2	043772-R	CSLOW	041426-R	ENDTST	046016-R	ERR9	046062-R	HLHIGH	041450-R	HLHIGH	041444-R	MRPLB	054126-R

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

AIDQR.TSK
QMAIN

MEMORY ALLOCATION MAP TKB
27-MAR-80

PAGE 3

Approved For Release 2005/07/11 : CIA-RDP85-00514R000200020001-3

MSTR1	041370-R	OUT1	046042-R	PRINT	041746-R	SDHIGH	041470-R	SQHIG	041460-R	TRTBL	041474-R	\$CEFI	005174
MSTR2	041372-R	PACK	045324-R	QXHIGH	041434-R	SDLOW	041472-R	SQLOW	041462-R	TRTBL2	041673-R	\$DIV	007146
MYSELF	041176-R	PDATA	044162-R	QXLOW	041436-R	SELPG	054614-R	SRHIGH	041464-R	TSKTCB	041202-R	\$DRDSE	017134
NEQLB	010000	PPCR	054532-R	RP	001000	SEQCS	053056-R	SRLow	041466-R	UNPK	045544-R	\$MUL	007116
OLDVEC	041204-R	PPLB	054462-R	RSPONT	041400-R	SEQMM	053560-R	STAT	041222-R	UPLIM	041402-R	\$TKTCB	004026
ONCE	000100	PPS	055330-R	RTNPT	041366-R	SPCR	055350-R	STOP	046036-R	WRTCS	053144-R		
OUT	002000	PRDATA	044354-R	SCAN	045114-R	SPS	055340-R	TIME	046022-R	WRTMM	053646-R		

AIDQR.TSK:7 MEMORY ALLOCATION MAP TKB
MRP 27-MAR-80

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

*** SEGMENT: MRP

R/W MEM LIMITS: 072014 072363 000350 00232.
DISK BLK LIMITS: 000075 000075 000001 00001.

MEMORY ALLOCATION SYNOPSIS:

SECTION	TITLE	IDENT	FILE
. BLK: (RW, I, LCL, REL, CON)	072014	000000	00000.
MRP: (RW, I, LCL, REL, CON)	072014	000226	00150.
	072014	000226	00150. MRP
\$\$\$ALVD: (RW, D, LCL, REL, CON)	072242	000120	00080.
\$\$\$RTS: (RW, I, GBL, REL, OVR)	071144	000002	00002.

GLOBAL SYMBOLS:

AT1	072322-R	G01	072332-R	MRP	072064-R	OF1	072342-R	RE1	072272-R	SS1	072352-R
CL1	072262-R	LD1	072242-R	MRPXX	072144-R	PR1	072252-R	RS1	072302-R	ST1	072312-R